



Bootstrap v 5.3

(material para Estudo)

Sumario

Breakpoint	3-5
Grid Template	5-12
Tipografia	13-14
Imagens e Carrousel	15-19
Formularios	20-22
Componentes Bootstrap	23-24
Modal	25
Classes Auxiliares do Bootstrap	26-28
Atributos BootStrap	28-30

Layout

Breakpoint

Breakpoint (pontos de interrupção)	Class infix (infixo da Classe)	Dimensions (Dimensões)
Small (Pequeno)	sm	≥576px
Medium (Médio)	md	≥768px
Large (Grande)	lg	≥992px
Extra large (Extra grande)	xl	≥1200px
Extra extra large (Extra extra grande)	xxl	≥1400px

Cada ponto de interrupção foi escolhido para conter confortavelmente contêineres cujas larguras são múltiplos de 12. Os pontos de interrupção também são representativos de um subconjunto de tamanhos de dispositivos comuns e dimensões de viewport — eles não visam especificamente cada caso de uso ou dispositivo. Em vez disso, os intervalos fornecem uma base forte e consistente para construir em quase qualquer dispositivo.

Ponto de interrupção único

Também há consultas de mídia e mixins para direcionar um único segmento de tamanhos de tela usando as larguras mínimas e máximas dos pontos de interrupção.

```
@include media-breakpoint-only(xs) { ... }  
@include media-breakpoint-only(sm) { ... }  
@include media-breakpoint-only(md) { ... }  
@include media-breakpoint-only(lg) { ... }  
@include media-breakpoint-only(xl) { ... }  
@include media-breakpoint-only(xxl) { ... }
```

Por exemplo, o `@include media-breakpoint-only(md) { ... }` resultado será:

```
@media (min-width: 768px) and (max-width: 991.98px) { ... }
```

Containers3(Recipientes)

Os contêineres são um bloco de construção fundamental do Bootstrap que contém, preenche e alinha seu conteúdo dentro de um determinado dispositivo ou janela de visualização.

Como eles funcionam

Os contêineres são o elemento de layout mais básico no Bootstrap e são **necessários ao usar nosso sistema de grade padrão**. Os contêineres são usados para conter, preencher e (às vezes) centralizar o conteúdo dentro deles. Embora os contêineres *possam* ser aninhados, a maioria dos layouts não requer um contêiner aninhado.

O Bootstrap vem com três contêineres diferentes:

- `.container`, que define um max-width em cada ponto de interrupção responsivo
- `.container-{breakpoint}`, que é width: 100% até o ponto de interrupção especificado
- `.container-fluid`, que está width: 100% em todos os pontos de interrupção

A tabela abaixo ilustra como cada contêiner max-width

se compara ao original `.container` `.container-fluid` em cada ponto de interrupção.

	Extra pequeno	Pequeno	Médio	Grande	Extra grande	XX-grande
	<576px	≥576px	≥768px	≥992px	≥1200px	≥1400px
<code>.container</code>	100%	540 px	720 px	960 px	1140 px	1320 px
<code>.container-sm</code>	100%	540 px	720 px	960 px	1140 px	1320 px
<code>.container-md</code>	100%	100%	720 px	960 px	1140 px	1320 px
<code>.container-lg</code>	100%	100%	100%	960 px	1140 px	1320 px
<code>.container-xl</code>	100%	100%	100%	100%	1140 px	1320 px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320 px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

Exemplos de Grid Bootstrap

Cinco níveis de grade

Existem cinco níveis no sistema de grade Bootstrap, um para cada faixa de dispositivos que suportamos. Cada nível começa em um tamanho mínimo de viewport e se aplica automaticamente a dispositivos maiores, a menos que seja substituído.

.col-4	.col-4	.col-4
.col-sm-4	.col-sm-4	.col-sm-4
.col-md-4	.col-md-4	.col-md-4
.col-lg-4	.col-lg-4	.col-lg-4
.col-xl-4	.col-xl-4	.col-xl-4
.col-xxl-4	.col-xxl-4	.col-xxl-4

Três colunas iguais

Obtenha três colunas de largura igual a partir de desktops e escalonando para desktops grandes. Em dispositivos móveis, tablets e abaixo, as colunas serão empilhadas automaticamente.

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

Três colunas iguais alternativa

Usando as classes `.row-cols-*`, você pode facilmente criar uma grade com colunas iguais.

.col child of .row-cols-md-3	.col child of .row-cols-md-3	.col child of .row-cols-md-3
------------------------------	------------------------------	------------------------------

Três colunas desiguais

Obtenha três colunas a partir de desktops e escalonando para desktops grandes com larguras variadas. Lembre-se, as colunas da grade devem somar doze para um único bloco horizontal. Mais do que isso, e as colunas começam a empilhar, independentemente do viewport.

.col-md-3	.col-md-6	.col-md-3
-----------	-----------	-----------

Duas colunas

Obtenha duas colunas a partir de desktops e escalonando para desktops grandes.

.col-md-8

.col-md-4

Largura total, coluna única

Nenhuma classe de grade é necessária para elementos de largura total.

Duas colunas com duas colunas aninhadas

Conforme a documentação, o aninhamento é fácil - basta colocar uma linha de colunas dentro de uma coluna existente. Isso fornece duas colunas a partir de desktops e escalonando para desktops grandes, com outras duas (larguras iguais) dentro da coluna maior.

Em tamanhos de dispositivos móveis, tablets e abaixo, essas colunas e suas colunas aninhadas serão empilhadas.

.col-md-8

.col-md-4

.col-md-6

.col-md-6

Misto: móvel e desktop

O sistema de grade do Bootstrap v5 possui seis níveis de classes: xs (extra pequeno, este infixo de classe não é usado), sm (pequeno), md (médio), lg (grande), xl (extra grande) e xxl (extra extra grande). Você pode usar quase qualquer combinação dessas classes para criar layouts mais dinâmicos e flexíveis.

Cada nível de classes escala para cima, o que significa que, se você planeja definir as mesmas larguras para md, lg, xl e xxl, você só precisa especificar md.

.col-md-8

.col-6.col-md-4

.col-6.col-md-4

.col-6.col-md-4

.col-6.col-md-4

.col-6

.col-6

Gutter

Com as classes .gx-*, os gutters horizontais podem ser ajustados.

<code>.col with .gx-4 gutters</code>	<code>.col with .gx-4 gutters</code>	<code>.col with .gx-4 gutters</code>
<code>.col with .gx-4 gutters</code>	<code>.col with .gx-4 gutters</code>	<code>.col with .gx-4 gutters</code>

Use as classes `.gy-*` para controlar os gutters verticais.

<code>.col with .gy-4 gutters</code>	<code>.col with .gy-4 gutters</code>	<code>.col with .gy-4 gutters</code>
<code>.col with .gy-4 gutters</code>	<code>.col with .gy-4 gutters</code>	<code>.col with .gy-4 gutters</code>

Com as classes `.g-*`, os gutters em ambas as direções podem ser ajustados

<code>.col with .g-3 gutters</code>	<code>.col with .g-3 gutters</code>	<code>.col with .g-3 gutters</code>
<code>.col with .g-3 gutters</code>	<code>.col with .g-3 gutters</code>	<code>.col with .g-3 gutters</code>

Largura igual

Por exemplo, aqui estão dois layouts de grade que se aplicam a todos os dispositivos e viewports, de xsa a xxl. Adicione qualquer número de classes sem unidade para cada ponto de interrupção que você precisar e cada coluna terá a mesma largura.

1 de 2

2 de 2

1 de 3

2 de 3

3 de 3

HTML -> ⚡ 📋

```
<div class="container text-center">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

Conteúdo de largura variável

Use `col-{breakpoint}-autoclasses` para dimensionar colunas com base na largura natural do seu conteúdo.

1 de 3

Conteúdo de largura variável

3 de 3

1 de 3

Conteúdo de largura variável

3 de 3

HTML -> ⚡ 📋

```
<div class="container text-center">
  <div class="row justify-content-md-center">
    <div class="col col-lg-2">
      1 of 3
    </div>
    <div class="col-md-auto">
      Variable width content
    </div>
    <div class="col col-lg-2">
      3 of 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-md-auto">
      Variable width content
    </div>
    <div class="col col-lg-2">
      3 of 3
    </div>
  </div>
</div>
```

Alinhamento vertical

Altere o alinhamento vertical com qualquer uma das align-items-*classes responsivas.

Uma das três colunas

Uma das três colunas

Uma das três colunas

HTML ->



```
<div class="container text-center">
  <div class="row align-items-start">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
</div>
```

As classes podem ser usadas junto com utilitários para criar imagens flutuantes responsivas. Certifique-se de envolver o conteúdo em um [.clearfix](#) wrapper para limpar o float se o texto for mais curto.

Um parágrafo de texto de espaço reservado. Estamos usando-o aqui para mostrar o uso da classe `clearfix`. Estamos adicionando algumas frases sem sentido aqui para demonstrar como as colunas interagem aqui com a imagem flutuante.

Como você pode ver, os parágrafos envolvem graciosamente a imagem flutuante. Agora imagine como isso ficaria com algum conteúdo real aqui, em vez de apenas esse texto de espaço reservado chato que continua e continua, mas na verdade não transmite nenhuma informação tangível. Ele simplesmente ocupa espaço e não deveria ser lido.

E ainda assim, aqui está você, ainda perseverando na leitura deste texto de espaço reservado, esperando por mais alguns insights, ou algum easter egg escondido de conteúdo. Uma piada, talvez. Infelizmente, não há nada disso aqui.

Responsive floated image

HTML->



```
<div class="clearfix">
  

  <p>
    A paragraph of placeholder text. We're using it here to show the use of the clearfix class. We'
  </p>

  <p>
    As you can see the paragraphs gracefully wrap around the floated image. Now imagine how this wo
  </p>

  <p>
    And yet, here you are, still persevering in reading this placeholder text, hoping for some more
  </p>
</div>
```

Responsivo

Use classes responsivas para ajustar seu layout em viewports. Aqui, começamos com duas colunas nas viewports mais estreitas e, em seguida, crescemos para três colunas em viewports médias e acima.

.g-col-6 .g-col-md-4

.g-col-6 .g-col-md-4

.g-col-6 .g-col-md-4

HTML-> ⚡ 📋

```
<div class="grid text-center">
  <div class="g-col-6 g-col-md-4">.g-col-6 .g-col-md-4</div>
  <div class="g-col-6 g-col-md-4">.g-col-6 .g-col-md-4</div>
  <div class="g-col-6 g-col-md-4">.g-col-6 .g-col-md-4</div>
</div>
```

Compare isso com este layout de duas colunas em todas as janelas de visualização.

.g-col-6

.g-col-6

HTML-> ⚡ 📋

```
<div class="grid text-center">
  <div class="g-col-6">.g-col-6</div>
  <div class="g-col-6">.g-col-6</div>
</div>
```

Tipografia:

Personalizando títulos#

Use as classes utilitárias incluídas para recriar o pequeno texto do título secundário do Bootstrap 3.

Cabeçalho de exibição sofisticado com texto secundário desbotado

HTML ->



```
<h3>
  Fancy display heading
  <small class="text-body-secondary">With faded secondary text</small>
</h3>
```

Liderar

Faça um parágrafo se destacar adicionando .lead.

Exibir títulos

Elementos de título tradicionais são projetados para funcionar melhor no conteúdo principal da sua página. Quando você precisa que um título se destaque, considere usar um **título de exibição** — um estilo de título maior e um pouco mais opinativo.

Exibir 1

Exibição 2

Exibir 3

Exibir 4

Exibir 5

Exibir 6

```
<h1 class="display-1">Display 1</h1>  
<h1 class="display-2">Display 2</h1>  
<h1 class="display-3">Display 3</h1>  
<h1 class="display-4">Display 4</h1>  
<h1 class="display-5">Display 5</h1>  
<h1 class="display-6">Display 6</h1>
```



Imagens:

O Bootstrap oferece um conjunto de classes CSS para formatar e manipular imagens de forma rápida e eficiente. Essas classes permitem que você torne suas imagens responsivas, adicione estilos básicos e integre-as perfeitamente em seus layouts.

Classes Principais:

- **.img-fluid:** Torna a imagem responsiva, ajustando seu tamanho para se encaixar perfeitamente em seu contêiner pai.

HTML

```

```

- **.img-thumbnail:** Adiciona um estilo de borda suave e um pouco de padding à imagem, criando o efeito de uma miniatura.

HTML

```

```

Outras Classes Úteis:

- **.rounded:** Arredonda os cantos da imagem.
- **.rounded-circle:** Cria uma imagem circular.
- **.rounded-top, .rounded-bottom, .rounded-left, .rounded-right:** Arredonda cantos específicos da imagem.
- **.img-top, .img-middle, .img-bottom:** Alinha verticalmente a imagem dentro de um elemento pai.

Alinhando imagens

Alinhe as imagens com as [classes auxiliares float](#) ou [classes de alinhamento de texto](#). `block`Imagens de nível `.-` podem ser centralizadas usando [a `.mx`-autoclasse de utilitário `margin`](#).



Tabelas ver no próprio site (<https://getbootstrap.com/docs/5.3/content/tables/>)

Classes Bootstrap para Criar Carousels

O Bootstrap oferece um componente chamado **carousel** (carrossel) que permite criar slideshows de imagens ou outros conteúdos, deslizando automaticamente ou manualmente. Para construir um carousel, você utilizará um conjunto específico de classes CSS.

Estrutura básica de um carousel:

HTML

```
<div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <h5>Primeiro slide</h5>
        <p>...</p>
      </div>
    </div>
  </div>
```

```

</div>
<button          class="carousel-control-prev"          type="button"          data-bs-
target="#carouselExampleControls" data-bs-slide="prev">
  <span class="carousel-control-prev-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Previous</span>
</button>
<button          class="carousel-control-next"          type="button"          data-bs-
target="#carouselExampleControls" data-bs-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Next</span>
</button>
</div>

```

Explicação das classes:

- **.carousel:** Classe principal que define o contêiner do carousel.
- **.carousel-inner:** Contém todos os itens do carousel.
- **.carousel-item:** Representa um único slide. A classe **.active** deve ser adicionada ao slide inicial.
- **.d-block w-100:** Garante que a imagem ocupe toda a largura do seu contêiner pai.
- **.carousel-caption:** Contém o conteúdo de texto (opcional) para cada slide.
- **.carousel-control-prev** e **.carousel-control-next:** Botões para navegar entre os slides.
- **.carousel-control-prev-icon** e **.carousel-control-next-icon:** Ícones dos botões de navegação.
- **.visually-hidden:** Texto oculto para acessibilidade.

Personalização:

- **Indicadores:** Adicione a classe **.carousel-indicators** para criar indicadores visuais abaixo do carousel.
- **Intervalos:** Use o atributo **data-bs-interval** para definir o tempo em milissegundos entre cada transição automática.
- **Controles:** Você pode personalizar a aparência dos botões de controle usando CSS.
- **Transições:** O Bootstrap utiliza transições CSS3 para criar um efeito suave de deslizamento.

Funcionalidades adicionais:

- **Responsividade:** O carousel se adapta automaticamente a diferentes tamanhos de tela.

- **Acessibilidade:** O carousel é projetado com foco em acessibilidade, com rótulos e atributos ARIA adequados.
- **Personalização:** Você pode personalizar o carousel usando CSS e JavaScript para criar designs personalizados.

Exemplo completo com indicadores:

HTML

```
<div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0"
class="active" aria-current="true" aria-label="Slide 1"></button>
  </div>
</div>
```


Observações:

- **JavaScript:** O carousel do Bootstrap depende de JavaScript para funcionar corretamente. Certifique-se de incluir o arquivo JavaScript do Bootstrap em seu projeto.
- **Responsividade:** Para garantir que o carousel se adapte bem a diferentes tamanhos de tela, utilize as classes de grid do Bootstrap para ajustar o layout.

Recursos adicionais:

- **Documentação oficial do Bootstrap:** [<https://getbootstrap.com/docs/5.3/components/carousel/>]

Figura



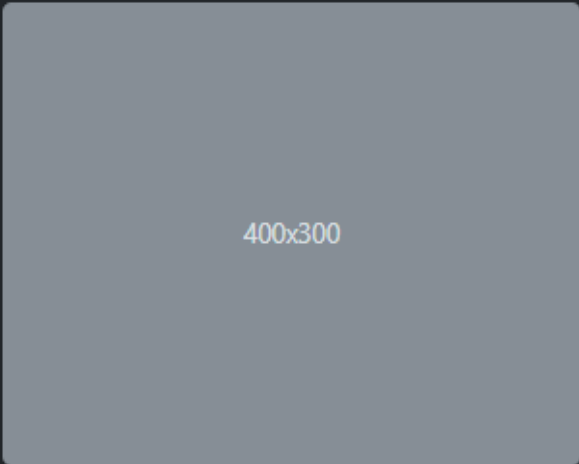
400x300

Uma legenda para a imagem acima.

HTML->

```
<figure class="figure">
  
  <figcaption class="figure-caption">A caption for the above image.</figcaption>
</figure>
```

Alinhar a legenda da figura é fácil com nossos [utilitários de texto](#) .



400x300

Uma legenda para a imagem acima.

HTML->

```
<figure class="figure">
  
  <figcaption class="figure-caption text-end">A caption for the above image.</figcaption>
</figure>
```

Formulários:

Visão geral

Os controles de formulário do Bootstrap expandem [nossos estilos de formulário Rebooted](#) com classes. Use essas classes para optar por suas exibições personalizadas para uma renderização mais consistente em navegadores e dispositivos.

Certifique-se de usar um `type` atributo apropriado em todas as entradas (por exemplo, `email` para endereço de e-mail ou `number` para informações numéricas) para aproveitar os novos controles de entrada, como verificação de e-mail, seleção de números e muito mais.

Aqui está um exemplo rápido para demonstrar os estilos de formulário do Bootstrap. Continue lendo para obter documentação sobre classes necessárias, layout de formulário e muito mais.

Endereço de email

Nunca compartilharemos seu e-mail com mais ninguém.

Senha

☐ Olha só pra mim

Enviar

HTML ->



```
<form>
  <div class="mb-3">
    <label for="exampleInputEmail1" class="form-label">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
    <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
  </div>
  <div class="mb-3">
    <label for="exampleInputPassword1" class="form-label">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1">
  </div>
  <div class="mb-3 form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Classes Fundamentais:

- **.form-control:** Aplica estilos básicos a elementos de entrada, como input, select e textarea, incluindo bordas, padding e foco.
- **.form-label:** Estiliza rótulos de formulário, alinhando-os verticalmente com os elementos de entrada.
- **.form-group:** Agrupa elementos de formulário relacionados, como rótulo e campo de entrada, para melhor organização.
- **.form-text:** Adiciona texto auxiliar abaixo de um elemento de entrada, como instruções ou mensagens de validação.

****Classes Adicionais para Personalização:** * ****

Tamanho:** ` .form-control-sm` , ` .form-control-lg`

Estado:** ` .is-valid` , ` .is-invalid` (para indicar se um campo é válido ou inválido) *

Layout * ` .form-row` (para criar linhas de formulários), ` .form-check` (para checkboxes e radio buttons)

Ajuda:** ` .form-text` (para adicionar texto auxiliar)

Formulários Horizontais:**

Para criar layouts de formulário horizontais, utilize as classes ` .form-row` e ` .col-*`.

Formulários Inline:

Para criar formulários mais compactos, use a classe .form-inline.

HTML

```
<form class="form-inline">
  <div class="form-group mb-2">
    </div>
  <div class="form-group mb-2">
    </div>
</form>
```

Componentes de Formulário:

O Bootstrap oferece componentes pré-construídos para elementos de formulário comuns, como:

- **Selects:** .form-select
- **Checkboxes:** .form-check-input, .form-check-label
- **Radio buttons:** .form-check-input, .form-check-label
- **File inputs:** .form-control-file
- **Switch:** .form-switch

Componentes Bootstrap: Classes Principais

1. Botões (Buttons)

- **Base:** `.btn` (para todos os botões)
- **Estilos:** `.btn-primary`, `.btn-secondary`, `.btn-success`, `.btn-danger`, `.btn-warning`, `.btn-info`, `.btn-light`, `.btn-dark` (para diferentes cores)
- **Tamanhos:** `.btn-lg`, `.btn-sm`, `.btn-xs` (para tamanhos diferentes)
- **Outlines:** `.btn-outline-*` (para botões com contorno)
- **Blocos:** `.btn-block` (para botões que ocupam toda a largura do contêiner pai)

2. Cards

- **Base:** `.card`
- **Corpo:** `.card-body`
- **Cabeçalho:** `.card-header`
- **Rodapé:** `.card-footer`
- **Imagem:** `.card-img-top`, `.card-img-bottom`
- **Título:** `.card-title`
- **Subtítulo:** `.card-subtitle`

3. Navegação (Nav)

- **Base:** `.nav`
- **Tabs:** `.nav-tabs`
- **Pills:** `.nav-pills`
- **Listas:** `.nav-list`
- **Itens:** `.nav-item`
- **Links:** `.nav-link`
- **Ativo:** `.active`

4. Listas (Lists)

- **Listas não ordenadas:** `<ul class="list-group">`
- **Listas ordenadas:** `<ol class="list-group">`
- **Itens:** `<li class="list-group-item">`

5. Formulários

- **Controles de entrada:** .form-control
- **Rótulos:** .form-label
- **Grupos:** .form-group
- **Checkboxes:** .form-check, .form-check-input, .form-check-label
- **Radio buttons:** .form-check, .form-check-input, .form-check-label
- **Selects:** .form-select
- **Textareas:** <textarea class="form-control">

6. Outros Componentes

- **Alertas:** .alert (com variações de cores)
- **Badges:** .badge
- **Progress Bars:** .progress, .progress-bar
- **Carousels:** .carousel
- **Dropdowns:** .dropdown
- **Modais:** .modal
- **Tooltips:** .tooltip
- **Popovers:** .popover

Observações:

- Esta é uma lista não exaustiva. O Bootstrap oferece muitos outros componentes e classes.
- As classes podem ser combinadas para criar estilos personalizados.

Para explorar mais:

- **Documentação Oficial:** <https://getbootstrap.com/docs/5.3/components/>

Modais no Bootstrap: Classes e Funcionamento

Modais no Bootstrap são caixas de diálogo que se sobrepõem ao conteúdo principal de uma página, geralmente usadas para exibir conteúdo adicional, alertas, formulários ou qualquer outro tipo de interação com o usuário.

Classes Principais:

- **.modal:** Define a caixa modal principal.
- **.modal-dialog:** Contém o conteúdo da modal.
- **.modal-content:** Contém o corpo da modal, incluindo o cabeçalho, corpo e rodapé.
- **.modal-header:** Contém o título da modal e o botão de fechar.
- **.modal-body:** Contém o conteúdo principal da modal.
- **.modal-footer:** Contém os botões de ação, como "Salvar" ou "Cancelar".

Classes Auxiliares do Bootstrap

O Bootstrap oferece um conjunto de classes utilitárias que fornecem maneiras rápidas de estilizar elementos sem a necessidade de escrever muito CSS. Essas classes "auxiliares" abrangem uma ampla gama de estilos, desde cores e espaçamento até visibilidade e muito mais.

Categorias Principais:

? Text:

- **Colors (Cores):** .text-primary, .text-secondary, .text-success, .text-danger, .text-warning, .text-info, .text-light, .text-dark, .text-body, .text-muted, .text-white, .text-black-50, .text-white-50
- **Weight (Peso):** .fw-bold, .fw-lighter, .fw-normal
- **Decoration (Decoração):** .text-decoration-none, .text-decoration-underline, .text-decoration-line-through
- **Transformation (Transformação):** .text-uppercase, .text-lowercase, .text-capitalize
- **Alignment (Alinhamento):** .text-start, .text-end, .text-center
- **Truncation (Trucamento):** .text-truncate

? Backgrounds (Fundos):

- **Colors:** .bg-primary, .bg-secondary, .bg-success, .bg-danger, .bg-warning, .bg-info, .bg-light, .bg-dark, .bg-white, .bg-transparent
- **Gradients:** Bootstrap offers classes for background gradients.

? Borders (Bordas):

- **Styles:** .border, .border-top, .border-end, .border-bottom, .border-start, .border-0
- **Colors:** .border-primary, .border-secondary, etc. (similar to text color classes)

? Spacing (Espaçamentos):

- **Margins:** .m-0, .m-1, .m-2, ..., .m-5, .m-auto, .mx-0, .my-0, .ms-0, .me-0, .mt-0, .mb-0 (similar for p for padding)
- **Negative Margins:** .m-n1, .m-n2, etc.

? Sizing (Tamanho):

- **Width:** .w-25, .w-50, .w-75, .w-100, .w-auto
- **Height:** .h-25, .h-50, .h-75, .h-100, .h-auto
- **Max Width/Height:** .mw-100, .mh-100

- **Viewport Width/Height:** .vw-100, .vh-100, .min-vw-100, .min-vh-100

? **Visibility (Visibilidade):**

- .d-none (display: none)
- .d-block (display: block)
- .d-inline (display: inline)
- .d-inline-block (display: inline-block)
- .d-flex (display: flex)
- **Breakpoint-specific visibility:** .d-sm-none, .d-md-block, etc.

? **Flexbox Utilities (Utilidades Flexbox):**

- **Alignment (alinhamento):** .justify-content-start, .justify-content-end, .justify-content-center, .justify-content-between, .justify-content-around, .align-items-start, .align-items-end, .align-items-center, .align-items-baseline, .align-items-stretch

? **justify-content-start:** Alinha os elementos flexíveis no início do eixo principal.

? **justify-content-end:** Alinha os elementos flexíveis no final do eixo principal.

? **justify-content-center:** Centra os elementos flexíveis ao longo do eixo principal.

? **justify-content-between:** Distribui os elementos flexíveis uniformemente ao longo do eixo principal, com o primeiro elemento no início e o último no final.

? **justify-content-around:** Distribui os elementos flexíveis uniformemente ao longo do eixo principal, com espaço igual em torno de cada item.

- **Direction (Direção):** .flex-row, .flex-column, .flex-row-reverse, .flex-column-reverse

Tipos de Classes de Utilidade:

- **Espaçamento:**
 - **Margens:** .m-0, .m-1, .m-2, ..., .m-5, .m-auto, .mx-0, .my-0, .ms-0, .me-0, .mt-0, .mb-0 (similar para p para padding)
 - **Margens Negativas:** .m-n1, .m-n2, etc.
- **Tamanho:**
 - **Largura:** .w-25, .w-50, .w-75, .w-100, .w-auto
 - **Altura:** .h-25, .h-50, .h-75, .h-100, .h-auto
 - **Largura Máxima/Altura:** .mw-100, .mh-100
 - **Largura/Altura do Viewport:** .vw-100, .vh-100, .min-vw-100, .min-vh-100
- **Visibilidade:**

- `.d-none` (`display: none`)
- `.d-block` (`display: block`)
- `.d-inline` (`display: inline`)
- `.d-inline-block` (`display: inline-block`)
- `.d-flex` (`display: flex`)
- **Visibilidade específica para breakpoint:** `.d-sm-none`, `.d-md-block`, etc.
- **Flexbox:**
 - **Alinhamento:** `.justify-content-start`, `.justify-content-end`, `.justify-content-center`, `.justify-content-between`, `.justify-content-around`, `.align-items-start`, `.align-items-end`, `.align-items-center`, `.align-items-baseline`, `.align-items-stretch`
 - **Direção:** `.flex-row`, `.flex-column`, `.flex-row-reverse`, `.flex-column-reverse`
- **Outras:**
 - **Cores:** `.text-primary`, `.bg-secondary`, etc.
 - **Bordas:** `.border`, `.border-top`, `.border-bottom`, etc.
 - **Arredondamento:** `.rounded`, `.rounded-circle`, etc.
 - **Sombra:** `.shadow`
 - **Opacidade:** `.opacity-50`, `.opacity-75`, etc

Atributos data-bs-* no Bootstrap

Os atributos data-bs-* são uma parte fundamental do Bootstrap, permitindo que você configure e controle componentes como modais, carousels, tooltips e outros, diretamente no HTML. Eles fornecem uma maneira concisa e intuitiva de interagir com esses componentes sem a necessidade de escrever muito JavaScript.

Como funcionam:

- **Prefixo data-bs-:** Indica que o atributo é específico do Bootstrap.
- **Nome do atributo:** Define a funcionalidade ou configuração que você deseja aplicar ao elemento.
- **Valor do atributo:** Especifica o valor para a configuração correspondente.

Exemplos de uso:

- **Modal:**

HTML

```
<button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
```

Abrir modal

```
</button>
```

- data-bs-toggle="modal": Indica que o botão deve abrir um modal.
- data-bs-target="#exampleModal": Especifica o ID do modal a ser aberto.

- **Carousel:**

HTML

```
<div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
```

```
</div>
```

- data-bs-ride="carousel": Inicia a animação do carousel automaticamente.

- **Collapse:**

HTML

```
<button class="btn btn-primary" data-bs-toggle="collapse" data-bs-target="#collapseExample">
```

Clique para expandir

```
</button>
```

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

</div>

- `data-bs-toggle="collapse"`: Indica que o botão deve mostrar ou ocultar o conteúdo.
- `data-bs-target="#collapseExample"`: Especifica o elemento a ser mostrado ou ocultado.

Atributos Comuns:

- **data-bs-toggle**: Especifica a ação a ser realizada, como mostrar/ocultar, abrir modal, etc.
- **data-bs-target**: Indica o elemento que será afetado pela ação.
- **data-bs-dismiss**: Fecha um componente, como um modal ou tooltip.
- **data-bs-interval**: Define o intervalo em milissegundos para transições automáticas (por exemplo, em carousels).
- **data-bs-parent**: Especifica um elemento pai para componentes como acordeões.
- **data-bs-placement**: Define o posicionamento de tooltips e popovers.

Listagem Completa:

Para uma lista completa e detalhada dos atributos `data-bs-*` disponíveis para cada componente do Bootstrap, consulte a documentação oficial: <https://getbootstrap.com/docs/5.3/>

Observações:

- **Personalização**: Os atributos `data-bs-*` permitem uma grande personalização dos componentes do Bootstrap sem a necessidade de escrever JavaScript personalizado.
- **Interação com JavaScript**: Embora os atributos `data-bs-*` simplifiquem a configuração, você ainda pode personalizar o comportamento dos componentes usando JavaScript.
- **Acessibilidade**: É importante usar os atributos ARIA junto com os atributos `data-bs-*` para garantir a acessibilidade dos seus componentes.

Conclusão:

Os atributos `data-bs-*` são uma ferramenta poderosa para criar interfaces de usuário interativas e dinâmicas com o Bootstrap. Ao entender como utilizá-los, você pode personalizar e configurar os componentes do Bootstrap de forma rápida e eficiente.

Gostaria de explorar algum componente específico ou tem alguma dúvida sobre os atributos `data-bs-*`?