

Bootstrap 4 (Biblioteca) → B7Web

Material de estudo desenvolvido por:

Daniel Teixeira Quadros

Sumário

Aula 01 – O Que É, Para Que Serve, Como Baixar.....	2
Aula 02 – Estruturando o Projeto.....	3
Aula 03 – Utilizando o console(inspecionar no navegador).....	3
Aula 04 – Grid – Introdução.....	4
Aula 05 – Grid – Coluna Com Largura e Nova Linha.....	5
Aula 06 – Grid – Ordenando Colunas.....	7
Aula 07 – Grid – JustifyContent e Alignitems.....	8
Aula 08 – Media Component.....	12
Aula 09 – Tipografia.....	14
Aula 10 – Imagens.....	16
Aula 11 – Tabelas.....	18
Aula 12 – Alertas e Avisos.....	21
Aula 13 – Botões e Grupos de Botões.....	25
Aula 14 – Carousel (Slideshow).....	27
Aula 15 - Collapse.....	30
Aula 16 – Accordion.....	31
Aula 17 – Dropdown.....	33
Aula 18 - Formulários.....	35
Aula 19 – input group.....	38
Aula 20 – list group.....	40
Aula 21 – modal.....	44
Aula 22 – navbar (Menu de Navegação).....	46
Aula 23 – Paginação.....	48
Aula 24 – tooltip (ferramenta dica).....	49
Aula 25 – Utilidades.....	51

Aula 01 – O Que É, Para Que Serve, Como Baixar

É a biblioteca mais popular de Front-End(Já vem pré pronta para criar seu site automaticamente responsivo :))

Obs: Ele foi criado para funcionar sem a necessidade do JavaScript(algumas funcionalidades, pelo menos as principais). Porém, outras coisas ela vai precisar do JavaScript.

A versão mais atual do Bootstrap encontrada hoje(19/04/2021) é a **versão 5.0.0 beta 3**. Aqui no curso, as aulas foram gravadas com a **versão 4.0.0**. e eu, resolvi utilizar a **versão 4.6**, pois estou dentro da versão 4, igual ao curso, porém achei por bem utilizar a 4.6, versão mais atualizada da 4. Obs.: Espero não me arrepender disso.

JQUERY → A versão mais atual do jQuery encontrada hoje(19/04/2021) é a **versão 3.6.0**. Aqui no curso, as aulas foram gravadas com a **versão 3.3.1**. e eu, resolvi utilizar a versão 3.6.0, pois estou dentro da versão 3, igual ao curso, porém achei por bem utilizar a 3.6.0, versão mais atualizada da 3. Obs.: Espero não me arrepender disso também.

O que utilizar do Bootstrap:

Ao Baixarmos o Bootstrap, encontramos diversos arquivos que vêm distribuídos em duas pastas: css e js
→ Da **pasta css**, copiamos para o nosso projeto o arquivo '**bootstrap.min.css**' que contém todas as css Bootstrap e com o código resumido(**.min**), ocupando menos espaço do que o arquivo 'bootstrap.css'.

Obs.: Por enquanto, ignoramos todos os outros arquivos.

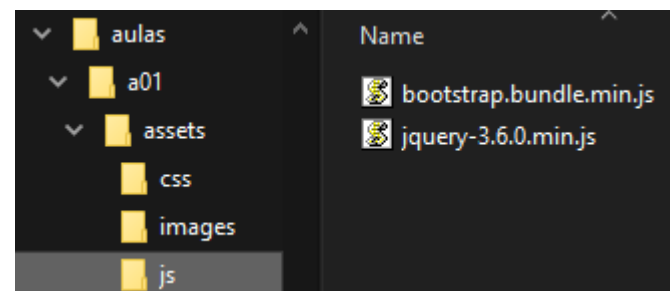
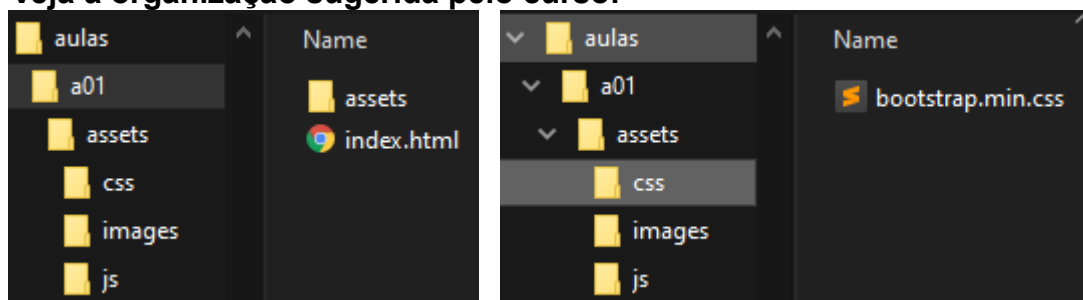
→ Da pasta js, copiamos para o nosso projeto o arquivo '**bootstrap.bundle.min.js**' que contém todo o JavaScript Bootstrap e com o código resumido(**.min**), ocupando menos espaço do que o arquivo 'bootstrap.bundle.js'.

Obs.: Por enquanto, ignoramos todos os outros arquivos.

O que utilizar do jQuery:

O jQuery é completamente baixado em apenas um arquivo .js, neste caso com o nome '**jquery-3.6.0.min.js**' e será colocado no nosso projeto junto com o arquivo .js do Bootstrap.

Veja a organização sugerida pelo curso:



Aula 02 – Estruturando o Projeto

ADICIONANDO O BOOTSTRAP E O JQUERY NO ARQUIVO HTML:

Entendendo, e adicionando parâmetro ao **<meta name="viewport"...>**

→ **viewport** diz para o navegador a forma que ele tem que renderizar o conteúdo na tela.

→ As propriedades do viewport estão no parâmetro **content**.

→ **width**(largura do site): a configuração "**device-width**" deixa o site na largura da tela do dispositivo que irá abrir o site(computador, tablet, celular).

→ **initial-scale**(zoom):a configuração '**1.0**' não adiciona zoom

Obs.: aqui adicionaremos mais uma propriedade que o VS Code não adiciona por padrão, que é a:

→ **shrink-to-fit=no** → esta propriedade diz para o navegador que ele não deve comprimir os elementos para que eles permaneçam na tela

Obs.: Neste caso, o navegador não irá comprimir os elementos, pois faremos isto utilizando o Bootstrap que encarregará de comprimi-los da forma correta. Veja o print:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
  <link rel="stylesheet" href="assets/css/bootstrap.min.css">
  <!-- link para adicionar o css do Bootstrap -->
  <title>a02 Bootstrap 4</title>
</head>
<body>
  <h1>Hello World!</h1>

  <script type="text/javascript" src="assets/js/jquery-3.6.0.min.js"></script>
  <script type="text/javascript" src="assets/js/bootstrap.bundle.min.js"></script>
  <!-- Obs.: Sempre carregar o jQuery antes do Bootstrap, conforme exemplo das duas linhas
  acima.-->
</body>
</html>
```

Aula 03 – Utilizando o console(inspecionar no navegador)

Aula 04 – Grid – Introdução

Para inserir conteúdo em uma estrutura feita utilizando o Bootstrap, é **altamente recomendável** que todo o conteúdo esteja dentro de **containers**.

container → Basicamente é uma <div> que determina até aonde o seu site vai(tamanho)

```
<div class="container" style="background-color: red;">
  <h1>Hello World!</h1>
</div>
```

Obs.: Aqui utilizamos o background-color, apenas para visualizarmos melhor o exemplo.

container-fluid → A diferença do container para o container-fluid é que o 'container-fluid' se expande até o limite da tela.

Obs.: Aqui não existe jeito certo ou errado, porém aqui no curso utilizaremos mais o container normal

```
<div class="container-fluid" style="background-color: yellow;">
  <h2>Título 2</h2>
</div>
```

Obs.: Aqui

também utilizamos o background-color, apenas para visualizarmos melhor o exemplo.

grid → Para criarmos uma grid (grade, ou tabela) precisamos fazer basicamente duas coisas:

- Primeiro vamos definir a linha (**row**).
- Depois, dentro de cada linha, definiremos quantas colunas teremos(**col**).

Obs. As colunas criadas se **auto ajustam sua largura** de acordo com o tamanho da tela dividindo o espaço proporcionalmente entre elas, graças a div container.

col-sm → Aqui, criaremos a coluna utilizando o parâmetro '**sm**'(small, ou pequeno), para que quando forem exibidas em uma tela pequena(celular) elas sejam empilhadas, para uma melhor visualização.

Obs.: Podemos definir o empilhamento a partir de outros tamanhos, neste caso, substituímos o sm, por outras siglas como o '**md**'(médio) ou '**lg**'(large, ou grande).

```
<div class="container"><!-- Conceito de grid-->
  <div class="row"><!-- row -> Linha-->
    <div class="col-sm" style="background-color: #FF0000;">...</div><!--
    <div class="col-sm" style="background-color: #00FF00">...</div>
    <div class="col-sm" style="background-color: red;">...</div>
  </div>
</div>
```

NAVEGADOR da aula 04:

Hello World!

Título 2

Aula 05 – Grid – Coluna Com Largura e Nova Linha

Para todos os exemplos abaixo, foi utilizado o seguinte CSS na <head> do HTML para melhor visualização das grids:

```
<style type="text/css">
    .row [class^=col-], .row .col {
        background-color: #DDD;
        border: 1px solid #CCC;
    }
</style>
```

Quebra de linha exemplo 1:

```
<div class="container"><!-- Primeiro teste: criar duas linhas de colunas-->
<!--Exemplo 1 utilizando uma div com class 'w-100'-->
<div class="row">
    <div class="col">...</div>
    <div class="col">...</div>
    <div class="col">...</div>
    <div class="w-100"></div><!-- Faz um quebra de linha -->
    <div class="col">...</div>
    <div class="col">...</div>
    <div class="col">...</div>
</div>
</div>
```

Quebra de linha exemplo2:

```
<div class="container">
    <!-- Exemplo 2, utilizando duas vezes a div com class row-->
    <div class="row">
        <div class="col">...</div>
        <div class="col">...</div>
        <div class="col">...</div>
    </div>
    <div class="row">
        <div class="col">...</div>
        <div class="col">...</div>
        <div class="col">...</div>
    </div>
</div>
```

Definindo tamanhos personalizados para cada coluna:

Obs.: Aqui, é importante entender como o Bootstrap faz a relação entre o tamanho das colunas.

→ O Bootstrap considera o tamanho total da tela como sendo o número 12. Se definirmos a largura de uma coluna com o número 2, por exemplo, restam a proporção 10 para o restante das colunas, ou seja, do tamanho máximo de 12, pegamos 2 para uma coluna específica e restaram 10 para a(s) outra(s). Veja o exemplo:

```

<!-- Definindo tamanhos personalizados para cada coluna -->
<div class="container">
  <div class="row">
    <div class="col">...</div>
    <div class="col-2">...</div><!-- Define proporção 2/12 -->
    <div class="col">...</div><!-- Quando há pelo menos uma coluna sem
o tamanho pré definido, ela, ou elas preencherão o restante do
espaço até o preenchimento total, ou seja, o 12/12-->
    <div class="w-100"></div>
    <!-- Outro exemplo. Aqui não será utilizada toda a tela, e sim
apenas 8/12 dela.-->
    <div class="col-2">...</div>
    <div class="col-2">...</div>
    <div class="col-4">...</div>
  </div>
</div>

```

Adicionando o **empilhamento** juntamente com o tamanho definido:

```

<div class="container">
  <!-- Também podemos, Extremamente recomendável, misturar com os parâmetros
de empilhamento para a adaptação inteligente entre os diversos tamanhos
de dispositivos-->
  <div class="row">
    <div class="col-sm">...</div>
    <div class="col-sm-2">...</div>
    <div class="col-sm-2">...</div>
  </div>
</div>

```

NAVEGADOR aula 05

...
...

...
...

...
...

...
...

Aula 06 – Grid – Ordenando Colunas

order-first e order-last:

```
<div class="container">
  <!-- Mudando a ordem das colunas (order-first e order-last)-->
  <div class="row">
    <div class="col order-last">Primeira</div>
    <!-- A classe order-last, neste exemplo, jogou a primeira coluna
    para a última posição da linha, empurrando as outras colunas
    para trás-->
    <div class="col">Segunda</div>
    <div class="col order-first">Terceira</div>
    <!-- A classe order-first, neste exemplo, jogou a terceira coluna
    para a primeira posição da linha, empurrando as outras colunas
    para frente-->
  </div>
</div>
```

Mudando a ordem das colunas de forma mais específica (com números):

Neste próximo exemplo, apesar de termos definido a quarta coluna na posição 1, ela apareceria na última posição, porque é a primeira, e neste caso a única, coluna que recebeu uma ordem. Concluindo, **Todas as colunas que não tem ordem definidas, virão na frente das que tem a ordem definida.**

```
<div class="container">
  <!-- Mudando a ordem das colunas de forma mais específica (com números)-->
  <div class="row">
    <div class="col">Primeira</div>
    <div class="col">Segunda</div>
    <div class="col">Terceira</div>
    <div class="col order-1">Quarta</div>
    <div class="col">Quinta</div>
  </div>
```

Ordenando todas as colunas:

```
<div class="container">
  <!-- Ordenando todas as colunas, conseguimos ter um melhor controle de posições-->
  <div class="row">
    <!-- No exemplo abaixo, colocamos a coluna Quarta na primeira posição-->
    <div class="col order-2">Primeira</div>
    <div class="col order-3">Segunda</div>
    <div class="col order-4">Terceira</div>
    <div class="col order-1">Quarta</div>
    <div class="col order-5">Quinta</div>
  </div>
</div>
```

NAVEGADOR aula 06

Terceira	Segunda	Primeira		
Primeira	Segunda	Terceira	Quinta	Quarta
Quarta	Primeira	Segunda	Terceira	Quinta

Aula 07 – Grid – JustifyContent e AlignItems

Obs.: Para os próximos exemplos utilizaremos as seguintes CSS:

```
<style type="text/css">
    .row [class^=col-], .row .col {
        background-color: #DDD;
        border: 1px solid #CCC;
    }
    .row{
        background-color: #FF9999;
        padding: 10px;
    }
</style>
```

ALINHAMENTO HORIZONTAL (justify-content)

Alinhamento a esquerda (start):

HTML:

```
<div class="container">
    <div class="row justify-content-start">
        <!--justify-content-start -> alinha as colunas a partir de esquerda(start)-->
        <div class="col-3">Primeira</div>
        <div class="col-3">Segunda</div>
        <div class="col-3">Terceira</div>
    </div>
</div>
```

NAVEGADOR:

Primeira	Segunda	Terceira
----------	---------	----------

Alinhamento centralizado(center):

HTML:

```
<div class="container">
    <div class="row justify-content-center">
        <!--justify-content-center -> alinha as colunas a partir do centro(center)-->
        <div class="col-3">Primeira</div>
        <div class="col-3">Segunda</div>
        <div class="col-3">Terceira</div>
    </div>
</div>
```

NAVEGADOR:

Primeira	Segunda	Terceira
----------	---------	----------

Alinhamento a Direita(end):

HTML:

```
<div class="container">
    <div class="row justify-content-end">
        <!--justify-content-end -> alinha as colunas a partir da direita(end)-->
        <div class="col-3">Primeira</div>
        <div class="col-3">Segunda</div>
        <div class="col-3">Terceira</div>
    </div>
</div>
```

NAVEGADOR:

Primeira

Segunda

Terceira

Alinhamento justificado centralizado(around):

HTML:

```
<div class="container">
  <div class="row justify-content-around">
    <!--justify-content-around vai fazer com que elas se espalhem, porém, o espaço
    entre uma e outra seja igualmente distribuído, como na configuração
    'justificado'(around) em um texto-->
    <div class="col-3">Primeira</div>
    <div class="col-3">Segunda</div>
    <div class="col-3">Terceira</div>
  </div>
</div>
```

NAVEGADOR:

Primeira

Segunda

Terceira

Alinhamento Justificado(between):

HTML:

```
<div class="container">
  <div class="row justify-content-between">
    <!--justify-content-between vai fazer com que elas se espalhem, fazendo que o
    espaço entre elas seja o maior possível. Neste caso, jogando a primeira
    coluna bem para a esquerda, a última coluna bem para a direita e dividindo
    o espaço restante entre elas-->
    <div class="col-3">Primeira</div>
    <div class="col-3">Segunda</div>
    <div class="col-3">Terceira</div>
  </div>
</div>
```

NAVEGADOR:

Primeira

Segunda

Terceira

ALINHAMENTO VERTICAL (align-items):

Obs.: Aqui, os parâmetro se repetem, porém o alinhamento é aplicado verticalmente. Abaixo, o HTML com três exemplos, o 'start', o 'center' e o 'end':

```

<div class="container">
  <div class="row align-items-start"><!-- O comando align-items trabalha
    o alinhamento vertical-->
    <!--align-items-start -> alinha as linhas a partir do topo(start)-->
    <div class="col-3">Primeira</div>
    <div class="col-3">Segunda</div>
    <div class="col-3">Terceira</div>
  </div>
</div>
<br>
<div class="container">
  <div class="row align-items-center">
    <!--align-items-center -> alinha as linhas a partir do centro(center)-->
    <div class="col-3">Primeira</div>
    <div class="col-3">Segunda</div>
    <div class="col-3">Terceira</div>
  </div>
</div>
<br>
<div class="container">
  <div class="row align-items-end">
    <!--align-items-end -> alinha as linhas a partir de baixo(end)-->
    <div class="col-3">Primeira</div>
    <div class="col-3">Segunda</div>
    <div class="col-3">Terceira</div>
  </div>
</div>

```

Obs-1.: Ainda podemos utilizar os parâmetros 'around' e 'between' para o alinhamento vertical.

Obs-2.: Podemos utilizar as configurações de alinhamento horizontal juntamente com as configurações de alinhamento vertical. Ex.: **<div class="row justify-content-center row align-items-center">**

Para este HTML acima, fizemos a seguinte modificação nas nossas CSS, **adicionamos** o parâmetro **height** nas duas **.row**.

```

<style type="text/css">
  .row [class^=col-], .row .col {
    background-color: #DDD;
    border: 1px solid #CCC;
    height: 50px;
  }
  .row{
    background-color: #FF9999;
    padding: 10px;
    height: 500px;
  }
</style>

```

Sobre o alinhamento vertical, trago apenas um exemplo no **navegador**, o alinhamento **center**

Primeira	Segunda	Terceira
----------	---------	----------

Aula 08 – Media Component

class="list-unstyled" → Tira todos os estilos de lista

class media → Classe utilizada pelo Facebook e outros para criar uma espécie de página ou campo de perfil.

class mr-3 avatar -> Este comando ainda não foi detalhado nesta aula

align-self-center → alinhar a si mesmo(self) e neste exemplo, utilizado o alinhamento 'center'. E aqui, podemos utilizar todos os recursos de alinhamentos aprendidos na aula passada

CSS:

```
<style type="text/css">
  .avatar{
    width: 50px;
    height: 50px;
  }
  .media{
    border-bottom: 1px solid #CCC;
    background-color: #EEEEEE;
  }
</style>
```

HTML:

```
<div class="container">
  <ul class="list-unstyled">
    <li class="media">
      
      <div class="media-body">
        <h5>Algum tipo de título</h5>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Odio maiores praesentium vitae enim qui amet veritatis? Sint molestias eligendi vero rem quod est quaerat consequuntur ex aspernatur, blanditiis facilis? Eveniet.</p>
      </div>
    </li>
    <li class="media">
      
      <div class="media-body">
        <h5>Algum tipo de título</h5>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Odio maiores praesentium vitae enim qui amet veritatis? Sint molestias eligendi vero rem quod est quaerat consequuntur ex aspernatur, blanditiis facilis? Eveniet.</p>
      </div>
    </li>
    <li class="media">
      
      <div class="media-body">
        <h5>Algum tipo de título</h5>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Odio maiores praesentium vitae enim qui amet veritatis? Sint molestias eligendi vero rem quod est quaerat consequuntur ex aspernatur, blanditiis facilis? Eveniet.</p>
      </div>
    </li>
  </ul>
</div>
```

NAVEGADOR:



Algum tipo de título

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Odio maiores praesentium vitae enim qui amet veritatis? Sint molestias eligendi vero rem quod est quaerat consequuntur ex aspernatur, blanditiis facilis? Eveniet.



Algum tipo de título

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Odio maiores praesentium vitae enim qui amet veritatis? Sint molestias eligendi vero rem quod est quaerat consequuntur ex aspernatur, blanditiis facilis? Eveniet.



Algum tipo de título

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Odio maiores praesentium vitae enim qui amet veritatis? Sint molestias eligendi vero rem quod est quaerat consequuntur ex aspernatur, blanditiis facilis? Eveniet.

Aula 09 – Tipografia

O Bootstrap traz a criação da **class h1** que podemos aplicar em outras tags:

```
<div class="h1">Algum Título</div>
```

class **text-muted**(outra classe do Bootstrap) → Desbota a cor do texto em que essa classe foi aplicada

Ex.:

```
<small class="text-muted">
```

class **mark** → marca o texto que estiver entre as tags em que essa classe for aplicada

Ex.:

```
<span class="mark"> adipisicing elit.</span>
```

class **text-right** → classe de alinhamento do Bootstrap. Aqui, podemos também usar outros recursos de alinhamento como **text-center** e **text-left**. Ex.:

```
<div class="teste text-right">Algum texto</div>
```

class **list-inline** → classe do Bootstrap que limpa as configurações padrões de listas carregadas pelo navegador. **Após adicionar a classe na , é necessário aplicá-la em cada adicionando o parâmetro 'item'**. Ex.:

```
<li class="list-inline-item">Item 1</li>
```

CSS usado na aula 09:

```
<style>
  .teste{
    width: 300px;
    height: 300px;
    border: 1px solid black;
  }
</style>
```

HTML Completo da aula 09

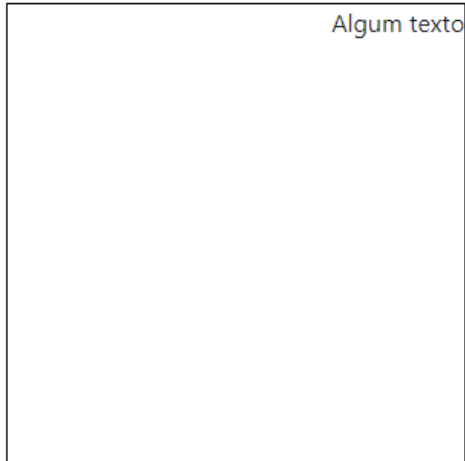
```
<div class="container">
  <div class="h1">Algum Título</div>
  <h3>Usuários<small class="text-muted">Adicionar usuários</small></h3>
  <p>Lorem ipsum dolor sit, amet <span class="mark"> adipisicing elit.</span> Accusantium ipsam deserunt labore, minima similique atque corporis, illo a esse dolor, delectus qui odio nesciunt amet repudiandae. Nam nostrum doloribus optio!</p>
  <div class="teste text-right">Algum texto</div>
  <div>
    <ul class="list-inline">
      <li class="list-inline-item">Item 1</li>
      <li class="list-inline-item">Item 2</li>
      <li class="list-inline-item">Item 3</li>
    </ul>
  </div>
</div>
```

NAVEGADOR:

Algum Título

UsuáriosAdicionar usuários

Lorem ipsum dolor sit, amet adipisicing elit. Accusantium ipsam deserunt labore, minima similique atque corporis, illo a esse dolor, delectus nesciunt amet repudiandae. Nam nostrum doloribus optio!



Algum texto

Item 1 Item 2 Item 3



Aula 10 – Imagens

img-fluid → Classe do Bootstrap que torna a imagem responsiva importante utilizar esta classe em todas as imagens para que elas fiquem responsivas.

HTML:

```
<div id="img1">
  
</div>
```

img-thumbnail → Classe Bootstrap que cria miniatura de uma imagem. Este efeito não altera as dimensões da imagem, apenas cria um efeito de moldura.

HTML: ``

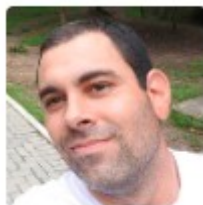
NAVEGADOR:



rounded → Classe Bootstrap que deixa a foto arredondada nos cantos.

HTML: ``

NAVEGADOR:



Observações.:

- Podemos utilizar essas classes do Bootstrap em **conjunto** umas com as outras. exemplo: `class="rounded img-fluid"`

- Ainda podemos utilizar configurações de alinhamento como por exemplo, `float-right` -> `class="rounded img-fluid float-right"`

figure → Classe 'figure' do Bootstrap permite que utilizemos Legenda em uma imagem. Esta legenda ficará abaixo da imagem.

Obs.: Para utilizarmos a classe 'figure', utilizamos a tag `<figure>` do HTML.

HTML:

```
<figure class="figure">
  
  <figcaption class="figure-caption text-right">Dog da Praia</figcaption>
</figure>
```

figure-img → Aqui, na tag `` utilizamos a classe `figure-img` para termos o efeito desejado.

figure-caption → Utilizamos a classe `figure-caption` para escrever a legenda que queremos adicionar. Neste exemplo adicionamos também o alinhamento do texto a direita na imagem.

NAVEGADOR:



Dog da Praia

Aula 11 – Tabelas

Tags <table>, <tr> e <th> → Tags HTML para criação de tabela. '<table>' cria a tabela, '<tr>' cria linha e '<th>' cria coluna.
HTML:

```
<table>
|
|   <tr>
|   |   <th>Item</th>
|   |
|   </tr>
|
</table>
```

table → Classe do Bootstrap que trabalha com a aparência da tabela.

HTML: `<table class="table">`

Obs.: Todas as configurações disponíveis para a classe 'table' podem ser utilizadas juntas.

table-dark → Inverte as cores da tabela

HTML: `<table class="table table-dark">`

NAVEGADOR:

#	Nome	Sobrenome	Idade
1	Daniel	Quadros	40
2	Elisiane	da Cruz	38
3	Gabrielle	Quadros	7

table-striped → Classe que alterna a coloração de cada linha da tabela para facilitar a visualização das informações de uma determinada linha (cor sim, cor não).

HTML: `<table class="table table-striped">`

NAVEGADOR:

#	Nome	Sobrenome	Idade
1	Daniel	Quadros	40
2	Elisiane	da Cruz	38
3	Gabrielle	Quadros	7

table-bordered → Esta classe adiciona uma borda em toda a nossa tabela.

HTML: `<table class="table table-bordered">`

NAVEGADOR:

#	Nome	Sobrenome	Idade
1	Daniel	Quadros	40
2	Elisiane	da Cruz	38
3	Gabrielle	Quadros	7

table-hover → Destaca a linha em que o cursor do mouse passar sobre.

HTML: `<table class="table table-hover">`

NAVEGADOR:

#	Nome	Sobrenome	Idade
1	Daniel	Quadros	40
2	Elisiane	da Cruz	38
3	Gabrielle	Quadros	7

thead → Tag utilizada para destacar o cabeçalho da tabela. Apenas essa tag sozinha, não oferece quase nenhum destaque, para termos um resultado mais expressivo, podemos utilizar algumas classes do Bootstrap como class="**thead-dark**" ou class="**thead-light**".

tbody → Tag utilizada para definir o restante da tabela, ou seja, o corpo.

HTML:

```
<table class="table">
  <thead class="thead-dark">
    <tr>
      <th>#</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>1</th>
    </tr>
```

NAVEGADOR:

#	Nome	Sobrenome	Idade
1	Daniel	Quadros	40
2	Elisiane	da Cruz	38
3	Gabrielle	Quadros	7

table-sm → Esta classe reduz o espaçamento entre os conteúdos.

HTML: `<table class="table table-sm">`

NAVEGADOR:

#	Nome	Sobrenome	Idade
1	Daniel	Quadros	40
2	Elisiane	da Cruz	38
3	Gabrielle	Quadros	7

table-responsive → Está classe trabalha em uma tabela que já é responsiva trazendo um recurso de criação de uma barra de rolagem horizontal, caso a tabela seja muito larga, a fim de facilitar o entendimento da leitura dos itens desta tabela.

HTML:

```
<div class="table-responsive">
  <table class="table">
    <thead class="thead-dark">
      <tr>
        <th>#</th>
```

NAVEGADOR:

#	Nome	Sobrenome	Idade
1	Daniel	Quadros	40
2	Elisiane	da Cruz	38
3	Gabrielle	Quadros	7

Aula 12 – Alertas e Avisos

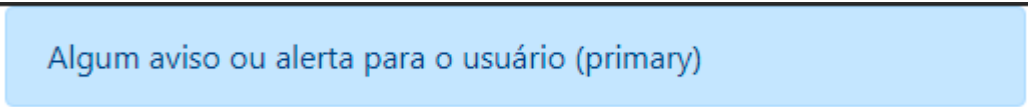
alert - Classe do Bootstrap que gera uma alerta. Essa classe pode ser utilizada sozinha, ou podemos complementar com o tipo de alerta que queremos utilizar.

alert-primary → Classe complementar da classe 'alert'. Trás o tom azul para o alerta.

HTML:

```
<div class="alert alert-primary" role="alert">
  |   Algum aviso ou alerta para o usuário (primary)
</div>
```

NAVEGADOR:



Algum aviso ou alerta para o usuário (primary)

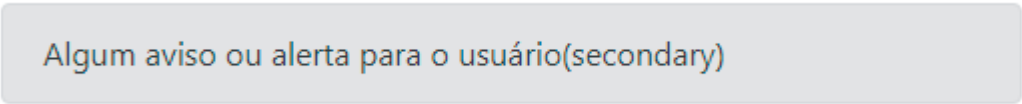
role="alert" → Não é obrigatório o uso desta role. Sua função é apenas informar ao JavaScript do Bootstrap que esta <div> se trata de um alerta.

alert-secondary → Esta classe também emite um alerta, porém, um pouco menos destacado, comparando com o alert-primary. Trás o tom cinza para o alerta

HTML:

```
<div class="alert alert-secondary" role="alert">
  |   Algum aviso ou alerta para o usuário(secondary)
</div>
```

NAVEGADOR:



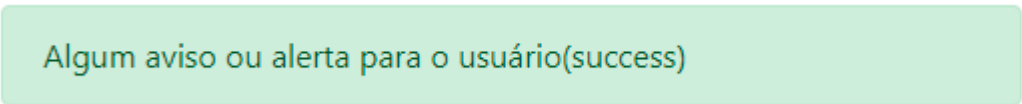
Algum aviso ou alerta para o usuário(secondary)

alert-success - Esta classe trás a ideia de aprovação. Trás o tom verde para o alerta.

HTML:

```
<div class="alert alert-success" role="alert">
  |   Algum aviso ou alerta para o usuário(success)
</div>
```

NAVEGADOR:

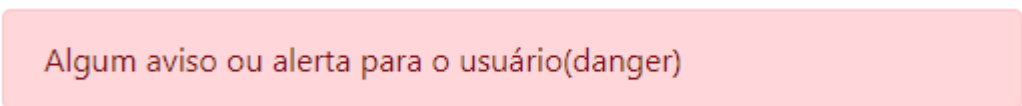


Algum aviso ou alerta para o usuário(success)

alert-danger - Como o próprio nome diz, dá uma idéia de perigo, ou reprovação. Trás o tom vermelho para o alerta.

HTML:

```
<div class="alert alert-danger" role="alert">
  |   Algum aviso ou alerta para o usuário(danger)
</div>
```



Algum aviso ou alerta para o usuário(danger)

NAVEGADOR:

alert-warning - Esta classe dá uma idéia de alerta, ou atenção. Trás o tom amarelo para o alerta.
HTML:

```
<div class="alert alert-warning" role="alert">
  |   Alguns avisos ou alertas para o usuário(warning)
</div>
```

NAVEGADOR:

Alguns avisos ou alertas para o usuário(warning)

alert-info - Classe que pode ser utilizada para apenas trazer uma informação. Trás o tom azul-claro para o alerta.
HTML:

```
<div class="alert alert-info" role="alert">
  |   Alguns avisos ou alertas para o usuário(info)
</div>
```

NAVEGADOR:

Alguns avisos ou alertas para o usuário(info)

alert-light - Esta classe pode ser utilizada em telas mais escuras. Trás o tom branco para o alerta.
HTML:

```
<div class="alert alert-light" role="alert">
  |   Alguns avisos ou alertas para o usuário()light
</div>
```

NAVEGADOR:

Alguns avisos ou alertas para o usuário()light

alert-dark - Esta classe pode ser utilizada em telas mais claras. Trás um tom cinza um pouco mais escuro do que ao cinza do 'alert-secondary' para o alerta.
HTML:

```
<div class="alert alert-dark" role="alert">
  |   Alguns avisos ou alertas para o usuário(dark)
</div>
```

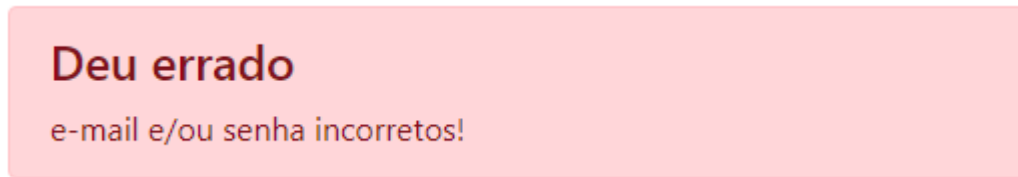
NAVEGADOR:

Alguns avisos ou alertas para o usuário(dark)

alert-heading - Trás característica de título, Destaque textual.
HTML:

```
<div class="alert alert-danger" role="alert">
  <h4 class="alert-heading">Deu errado</h4>
  e-mail e/ou senha incorretos!
</div>
```

NAVEGADOR:



alert-dismissible - Classe utilizada para adicionar um botão para o usuário fechar a mensagem de alerta. Obs.: Esta classe apenas, não cria o botão de fechamento, ela apenas estiliza o botão.

close - Ajuda na formatação do botão fechar.

data-dismiss="alert" - Diz para o Bootstrap que tipo de ação vai ocorrer. Neste exemplo, o 'dismiss' diz que o 'alert' deve desaparecer.

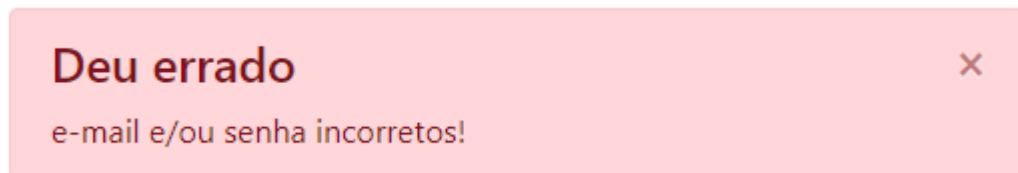
aria-label="Fechar" - Não é obrigatório(e pelo que eu vi, também não foi necessário) a sua utilização. Não foi comentado nada sobre qual é a sua função.

× - Esta linha que gerou o nosso botão para fechar, sendo que o '×' que criou o 'x'.

HTML:

```
<div class="alert alert-danger alert-dismissible"
role="alert">
  <h4 class="alert-heading">Deu errado</h4>
  e-mail e/ou senha incorretos!
  <button class="close" data-dismiss="alert"
  aria-label="Fechar">
    <span aria-hidden="true">&times;</span>
  </button>
</div>
```

NAVEGADOR:



fade - Faz com que, no fechamento do alerta, ele não desapareça momentaneamente. Ele dá um efeito de 'ir desaparecendo'. Para que esta classe gere o efeito desejado, deve ser combinada com a classe 'show'.

show - Classe que contribui para que o efeito da classe **fade** seja visível.

HTML:

```
<div class="alert alert-danger alert-dismissible fade
show" role="alert">
  <h4 class="alert-heading">Deu errado</h4>
  e-mail e/ou senha incorretos!
  <button class="close" data-dismiss="alert"
aria-label="Fechar">
    <span aria-hidden="true">&times;</span>
  </button>
</div>
```

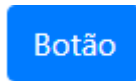
Aula 13 – Botões e Grupos de Botões

btn - Classe Bootstrap utilizada para padronização de botões, independentemente da tag que foi utilizada para a sua criação, ela remove qualquer tipo de configuração pré definida.

btn-primary - Classe que trás uma aparência de botão com bordas arredondadas e de cor azul.

HTML: `Botão`

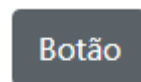
NAVEGADOR:



btn-secondary - Classe que trás uma aparência de botão com bordas arredondadas e de cor cinza.

HTML: `Botão`

NAVEGADOR:

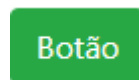


btn-success - Classe que trás uma aparência de botão com bordas arredondadas e de cor verde.

Obs.: Aqui, assim como os alertas, que vimos na aula passada, temos todas as outras variações: '-danger', '-warning', '-info', '-light', '-dark'.

HTML: `Botão`

NAVEGADOR:



btn-link - Essa classe trás uma aparência de link.

Obs.: com o btn-link, **podemos combinar as outras classes btn** para aplicarmos a estilização de cor e formato desejados.

HTML: `Botão`

NAVEGADOR:

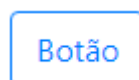
Botão

btn-outline-primary - Com a adição do 'outline', o botão fica apenas com a borda, ou seja sem a cor de fundo. A cor de fundo aparece apenas quando passamos o mouse sobre ele.

Obs.: a variação de estilo oferecido pelo 'outline', **pode ser combinada com todas as outras configurações** de botões vistas acima.

HTML: `Botão`

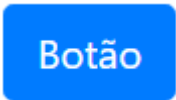
NAVEGADOR:



btn-lg - Classe que modifica o tamanho do botão. Neste caso, 'lg' ou seja, grande (large). Assim, também, temos a variável 'sm', ou seja pequena (small) para deixar o botão menor do que o padrão.

HTML: `<button class="btn btn-primary btn-lg">Botão</button>`

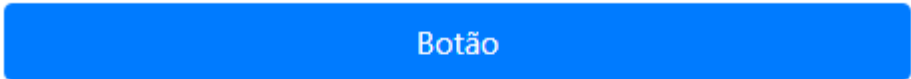
NAVEGADOR:



btn-block - Esta classe faz com que o botão tome conta de toda a largura disponível (responsivo).

HTML: `<input class="btn btn-primary btn-block" type="submit" value="Botão">`

NAVEGADOR:

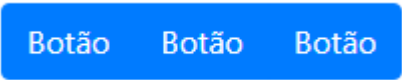


btn-group - Esta classe, reúne todos os botões que estão sob a mesma tag, neste exemplo, uma <div> e dá uma aparência de um bloco único.

HTML:

```
<div class="btn-group">
  <a class="btn btn-primary" href="#">Botão</a>
  <button class="btn btn-primary">Botão</button>
  <input class="btn btn-primary" type="submit" value="Botão">
</div>
```

NAVEGADOR:

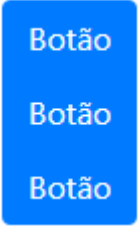


btn-group-vertical - Com a adição do 'vertical' podemos utilizar o btn-group de forma empilhada, ou seja, verticalmente.

HTML:

```
<div class="btn-group-vertical">
  <a class="btn btn-primary" href="#">Botão</a>
  <button class="btn btn-primary">Botão</button>
  <input class="btn btn-primary" type="submit" value="Botão">
</div>
```

NAVEGADOR:



Aula 14 – Carousel (Slideshow)

slide carousel - Classes do Bootstrap que geram o efeito de slideshow. Para este efeito, elas trabalham em conjunto.

slide - Define o tipo de processo, de transição que será realizado. É o slide que trás o efeito de deslizar, na mudança de itens.

HTML: `<div id="slideshowExemplo" class="slide carousel">`

carousel-indicators - Classe adicionada para criarmos os indicadores de slide. Precisamos utilizar esta estrutura HTML de `` e ``

HTML: `<ol class="carousel-indicators">`

data-target="#slideshowExemplo data-slide-to="0" - Este item é o representativo da foto número zero.

Assim como em uma array, começando do zero colocamos um número para cada item.

Assim como no carrocel, a classe active, define qual o primeiro indicador que aparecerá no carrocel.

HTML:

```
<li data-target="#slideshowExemplo" data-slide-to="0" class="active"></li>
<li data-target="#slideshowExemplo" data-slide-to="1"></li>
<li data-target="#slideshowExemplo" data-slide-to="2"></li>
<li data-target="#slideshowExemplo" data-slide-to="3"></li>
```

carousel-inner - Esta classe define o início da declaração dos itens internos do carrocel.

HTML: `<div class="carousel-inner">`

carousel-item - Utilizada para a declaração de cada um dos itens que farão parte do carousel.

active - Define qual será o primeiro elemento que irá aparecer na apresentação.

HTML: `<div class="carousel-item active">`

w-100 - Define que a largura de cada imagem ocupara 100% do espaço na apresentação do slideshow.

HTML: ``

Para a criação dos botões que alternam entre as imagens do slide, precisamos **criar um ID para a primeira <div>** desta estrutura, ou seja, a `<div>` que contém a classe 'slide carousel'. Para este exercício colocamos o ID 'slideshowExemplo'.

Após o fechamento da `<div>` que contém a class 'carousel-inner' criamos um link com o href iniciando com um '#' e sem colocar espaço, adicionamos o ID da `<div>` que contém a classe 'slide carousel' Ainda neste link, definimos o botão retroceder com o parâmetro 'data-slide="prev"' e o botão de avançar utilizará o parâmetro 'data-slide=next'.

carousel-control-prev - Esta classe executa o retrocesso do ítem.

HTML: ``

Dentro do link adicionamos um span

carousel-control-prev-icon - Classe que adiciona o botão para retroceder o slide.

HTML: ``

carousel-control-next - Esta classe executa o avanço do ítem.

HTML: ``

carousel-control-next-icon - Classe que adiciona o botão para avançar o slide.

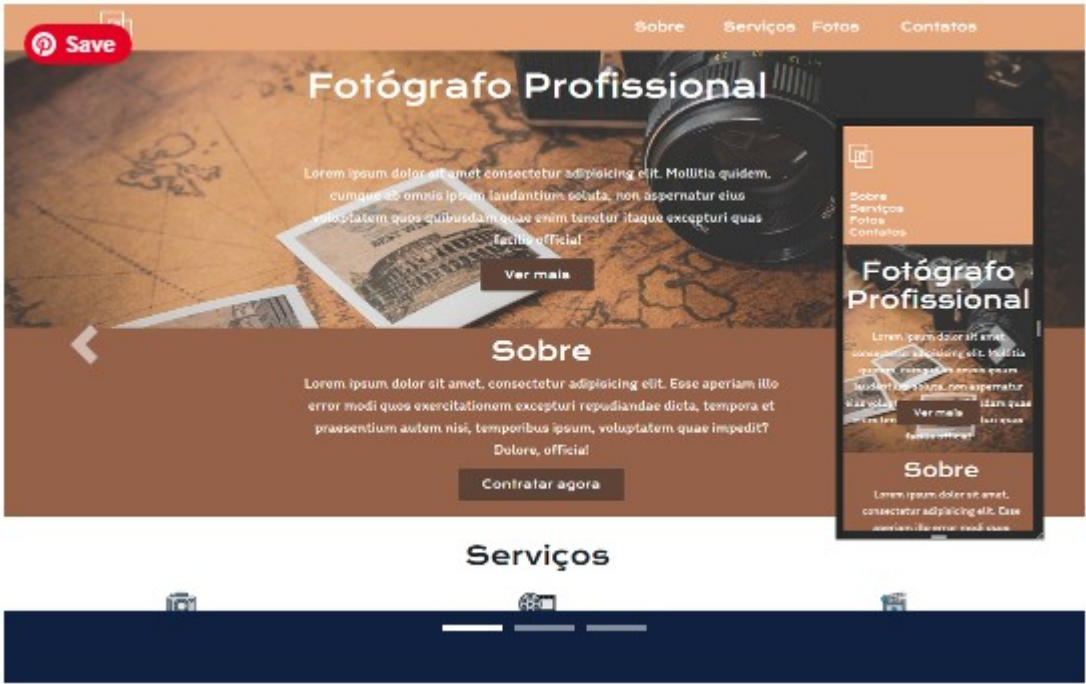
HTML: ``

Além destas possibilidades de configurações que fizemos no carrocel, na documentação do Bootstrap, existem diversas outras.

Abaixo o código completo desta aula (Aula 14 – Carousel):

```
<div class="container">
  <div id="slideshowExemplo" class="slide carousel">
    <ol class="carousel-indicators">
      <li data-target="#slideshowExemplo" data-slide-to="0" class="active"></li>
      <li data-target="#slideshowExemplo" data-slide-to="1"></li>
      <li data-target="#slideshowExemplo" data-slide-to="2"></li>
      <li data-target="#slideshowExemplo" data-slide-to="3"></li>
    </ol>
    <div class="carousel-inner">
      <div class="carousel-item active">
        
      </div>
      <div class="carousel-item">
        
      </div>
      <div class="carousel-item">
        
      </div>
      <div class="carousel-item">
        
      </div>
    </div>
    <a class="carousel-control-prev" href="#slideshowExemplo" data-slide="prev">
      <span class="carousel-control-prev-icon"></span>
    </a>
    <a class="carousel-control-next" href="#slideshowExemplo" data-slide="next">
      <span class="carousel-control-next-icon"></span>
    </a>
  </div>
</div>
```

NAVEGADOR, UTILIZEI O EXEMPLO DE CARROCEL DO MEU PORTFÓLIO, PARA UMA VISUALIZAÇÃO MELHOR:



Aula 15 - Collapse

collapse → Significa que teremos um botão, e ao clicarmos neste botão algum conteúdo que estava anteriormente escondido, vai aparecer na tela. Se clicarmos novamente, este conteúdo desaparece.

Obs.: A Tag que utilizaremos para aplicar esta classe, não tem importância. Neste exemplo utilizaremos a tag button.

data-toggle="collapse" -> Define o funcionamento deste processo. Neste exemplo, quando clicarmos neste botão, ele irá acionar o collapse em um outro item, que indicaremos através do data-target.

HTML:

```
<div class="container">
  <button class="btn btn-primary" data-toggle="collapse" data-target="#area">Mostrar/Esconder</button>
  <div class="collapse" id="area">
    |   O texto escondido que vai aparecer/desaparecer.
  </div>
</div>
```

NAVEGADOR:

Mostrar/Esconder

O texto escondido que vai aparecer/desaparecer.

Aula 16 – Accordion

card → Utilizado apenas para efeitos visuais.

HTML: `<div class="card">`

card-header → Cria o título do card.

card-body → Cria o corpo do card.

HTML: `<div class="card-body">`

aria-controls="c1" → Este comando é o que fará a associação entre este botão e a div de id c1.

HTML: `<button class="btn btn-link" data-toggle="collapse" data-target="#c1" aria-controls="c1">`

data-parent="accordion" → Utilizamos este comando para fazermos a ligação deste card com a div que colocamos o id="accordion". Este comando é importante, por que através dele, poderemos fazer a ligação entre todos os itens deste accordion.

Obs. 1: Todos itens que receberem o data-parent="accordion", se alternarão entre si.

Obs. 2: Aqui, utilizamos o show junto com o collapse, para que ele inicie aparecendo.

HTML: `<div id="c1" class="collapse show" data-parent="#accordion">`

ESTRUTURA COMPLETA NO HTML:

```
<div id="accordion">
  <div class="card">
    <div class="card-header">
      <button class="btn btn-link" data-toggle="collapse" data-target="#c1"
        aria-controls="c1">Título do card 1</button>
    </div>
    <div id="c1" class="collapse show" data-parent="#accordion">
      <div class="card-body">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit, explicabo
        cum architecto sequi aspernatur, beatae cumque voluptatum blanditiis quae
        earum vitae. Ipsam autem aliquid reiciendis ipsa voluptatum, maxime
        accusamus enim!
      </div>
    </div>
  </div>
  <div class="card">
    <div class="card-header">
      <button class="btn btn-link" data-toggle="collapse" data-target="#c2"
        aria-controls="c2">Título do card 2</button>
    </div>
    <div id="c2" class="collapse" data-parent="#accordion">
      <div class="card-body">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit, explicabo
        cum architecto sequi aspernatur, beatae cumque voluptatum blanditiis quae
        earum vitae. Ipsam autem aliquid reiciendis ipsa voluptatum, maxime
        accusamus enim!
      </div>
    </div>
  </div>
</div>
```

NAVEGADOR:

Título do card 1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Fugit, explicabo cum architecto sequi aspernatur, beatae cumque voluptatum blanditiis quae earum vitae. Ipsam autem aliquid reiciendis ipsa voluptatum, maxime accusamus enim!

Título do card 1

Título do card 1

Aula 17 – Dropdown

dropdown → Classe Bootstrap que cria um botão que ao clicar nele, ele abre um menu.

dropright → Podemos também utilizar esta classe para fazer com que o menu, ao invés de abrir para baixo, abra para a direita.

Obs.: Também temos as classes `dropleft` e `dropup`.

HTML: `<div class="dropdown dropright">`

dropdown-toggle - Esta classe adiciona uma seta no botão para indicar que se for clicado, algo será aberto.

data-toggle="dropdown" → Este comando faz com que ao clicar uma vez o menu irá abrir e ao clicar novamente o menu irá fechar.

HTML: ``

dropdown-menu → É com esta classe que criaremos todo o menu que irá abrir ao clicar no botão.

HTML: `<div class="dropdown-menu">`

dropdown-divider - Produz um efeito de divisão no menu, neste caso, não precisamos adicionar nenhum valor na div, apenas esta classe.

HTML: `<div class="dropdown-divider"></div>`

dropdown-item → Aprimora a estilização dos itens do menu.

HTML: ``

Obs.: Em algumas ocasiões, podemos precisar que o botão, funcione realmente como um botão, ou seja, que além da função de abrir um menu. Para isso utilizamos a classe **btn-group** juntamente com a classe `dropdown`.

HTML: `<div class="dropdown btn-group">`

dropdown-toggle-split - indica que existe um split (divisão) do botão.

``

HTML COMPLETO:

```
<div class="container">
  <div class="dropdown dropright">
    <a href="#" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">Editar</a>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="https://google.com.br">Google</a>
      <a class="dropdown-item" href="https://Facebook.com.br">Facebook</a>
      <div class="dropdown-divider"></div>
      <a class="dropdown-item" href="https://youtube.com.br">Youtube</a>
    </div>
  </div>
  <br>
  <div class="dropdown btn-group">
    <a href="https://instagram.com" class="btn btn-primary">Instagram</a>
    <a href="#" class="btn btn-primary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown"></a>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="https://google.com.br">Google</a>
      <a class="dropdown-item" href="https://Facebook.com.br">Facebook</a>
      <div class="dropdown-divider"></div>
      <a class="dropdown-item" href="https://youtube.com.br">Youtube</a>
    </div>
  </div>
</div>
```

Editar ▶

Instagram ▼

- Google
 - Facebook
 - Youtube
-

Aula 18 - Formulários

form-group - Classe Bootstrap para a estilização de formulário

HTML: `<div class="form-group">`

for="email" - HTML que vincula o título deste input com o próprio input

HTML: `<label for="email1">E-mail 1:</label>`

form-control - Classe Bootstrap de estilização para campo específico de formulário.

form-control-lg - Variação da classe form-control, que se agrupada junto a ela, altera o tamanho do campo. Neste caso, aumentando ligeiramente o espaço (**lg -> large, grande**). também podemos utilizar a variação sm para tornar um pouco menor que o padrão (**sm -> small, pequeno**).

HTML: `<input id="email1" type="email" name="email1" class="form-control form-control-lg">`

Aqui, utilizamos o **HTML value** para que o nosso input já venha preenchido com algum valor, porém, permitindo que o usuário edite este valor.

HTML: `<input id="email2" type="email" name="email2" class="form-control" value="danielqtquadros@hotmail.com">`

Neste exemplo, adicionamos a propriedade **readonly**, para que não seja possível a edição deste input.

HTML: `<input id="email3" type="email" name="email3" class="form-control" value="danielqtquadros@hotmail.com" readonly>`

form-control-plaintext - Esta variação de form-control torna o nosso input com estilo de texto.

HTML: `class="form-control-plaintext" value="danielqtquadros@hotmail.com" readonly>`

form-inline - Esta classe Bootstrap, aplicada no `<form>` estiliza o formulário horizontalmente.

HTML: `<form method="POST" class="form-inline">`

TAMBÉM PODEMOS MESCLAR A ORGANIZAÇÃO DO NOSSO FORMULÁRIO COM AS ESTILIZAÇÕES DE GRID:

form-row - Classe que mescla a estilização de formulário com a estilização de grid.

HTML: `<div class="form-row">`

HTML DOS TRÊS FORMULÁRIOS MONTADOS NESTE EXERCÍCIO COM OS RESPECTIVOS RESULTADOR NO NAVEGADOR:

FORMULÁRIO 1:

HTML:

```

<div class="container">
  <form method="POST">
    <div class="form-group">
      <label for="email1">E-mail 1:</label>
      <input id="email1" type="email" name="email1" class="form-control form-control-lg">
    </div>
    <div class="form-group">
      <label for="email2">E-mail 2:</label>
      <input id="email2" type="email" name="email2" class="form-control" value="danieltquadros@hotmail.com">
    </div>
    <div class="form-group">
      <label for="email3">E-mail 3:</label>
      <input id="email3" type="email" name="email3" class="form-control" value="danieltquadros@hotmail.com" readonly>
    </div>
    <div class="form-group">
      <label for="email4">E-mail 4:</label>
      <input id="email4" type="email" name="email4" class="form-control-plaintext" value="danieltquadros@hotmail.com" readonly>
    </div>
    <div class="form-group">
      <label for="senha">Senha:</label>
      <input id="senha" type="password" name="senha" class="form-control">
    </div>
    <div class="form-group">
      <input type="submit" value="Entrar" class="btn btn-primary">
    </div>
  </form>
  <hr>
</div>

```

NAVEGADO:

E-mail 1:

E-mail 2:

E-mail 3:

E-mail 4:

danieltquadros@hotmail.com

Senha:

Entrar

FORMULÁRIO

2:

HTML:

```

<div class="container">
  <form method="POST" class="form-inline">
    <div class="form-group">
      <input type="email" name="email" class="form-control" placeholder="E-mail">
    </div>
    <div class="form-group">
      <input type="password" name="senha2" class="form-control"
        placeholder="Senha">
    </div>
    <div class="form-group">
      <input type="submit" value="Entrar" class="btn btn-primary">
    </div>
  </form>
  <hr>
</div>

```

NAVEGADOR:

FORMULÁRIO 3:

HTML:

```

<div class="container">
  <form method="POST">
    <div class="form-row">
      <div class="col">
        <div class="form-group">
          <input type="text" name="nome" class="form-control"
            placeholder="Nome completo">
        </div>
      </div>
      <div class="col">
        <div class="form-group">
          <input type="email" name="email" class="form-control"
            placeholder="E-mail">
        </div>
      </div>
      <div class="col">
        <div class="form-group">
          <input type="password" name="senha2" class="form-control"
            placeholder="Senha">
        </div>
      </div>
    </div>
    <div class="form-row">
      <div class="col">
        <div class="form-group">
          <input type="submit" value="Entrar" class="btn btn-primary">
        </div>
      </div>
    </div>
  </form>
  <hr>
</div>

```

NAVEGADOR:

Aula 19 – input group

input-group → Classe Bootstrap para estilização de input.

```
<div class="input-group-prepend">
```

input-group-prepend → Classe que padroniza o **prefixo** do input, afim de que o usuário preencha apenas o restante do campo.

```
<div class="input-group-prepend">
```

input-group-text → Esta classe cria o texto com o texto complementar de forma que o usuário entenda qual é a informação que ele necessita complementar.

```
<span class="input-group-text">@</span>
```

input-group-sm → O 'sm' trás uma variação de tamanho, neste caso, para pequeno (sm). Também existe a variação para grande (**lg**).

```
<div class="input-group input-group-sm">
```

input-group-append - Classe que padroniza o sufixo do input, afim de que o usuário preencha apenas o restante do campo.

```
<div class="input-group-append">
```

Também podemos utilizar a variação de **prepend** e **append** em um mesmo input.

HTML COMPLETO:

```
<div class="container">
  <form method="POST">
    <!-- PREPEND-->
    <div class="input-group">
      <div class="input-group-prepend">
        <span class="input-group-text">@</span>
        <input type="text" name="instagram" class="form-control" placeholder="Usuário">
      </div>
    </div>
    <br>
    <!-- APPEND-->
    <div class="input-group input-group-sm">
      <input type="email" name="email" class="form-control" placeholder="Seu e-mail aqui">
      <div class="input-group-append">
        <span class="input-group-text">@hotmail.com</span>
      </div>
    </div>
    <br>
    <!-- PREPEND E APPEND-->
    <div class="input-group">
      <div class="input-group-prepend">
        <span class="input-group-text">R$</span>
        <input type="number" name="valor" class="form-control" placeholder="Valor">
        <div class="input-group-append">
          <span class="input-group-text">,00</span>
        </div>
      </div>
    </div>
  </form>
</div>
```

NAVEGADOR:

@

Usuário

Seu e-mail aqui

@hotmail.com

R\$

Valor

,00



Aula 20 – list group

list group → Lista agrupada, pode fazer com diversas tag como , ou com divs mesmo.

Obs.: Estas divs aqui, foram criadas apenas para ajudar a visualização do exercício.

```
<div class="container">  
<div style="width: 600px; margin: 20px;">
```

list-group → Classe Bootstrap que aplica a estilização de lista.

list-group-item - Classe Bootstrap que define cada item desta lista.

active - Pode ser utilizado para "ativar", ou destacar o ítem em que for aplicado.

HTML:

```
<ul class="list-group">  
  <li class="list-group-item">Item 1</li>  
  <li class="list-group-item">Item 2</li>  
  <li class="list-group-item">Item 3</li>  
</ul>
```

NAVEGADOR:

Item 1
Item 2
Item 3

active → Pode ser utilizado para "ativar", ou destacar o ítem em que for aplicado.

disabled → "desativa", ou seja, deixa o ítem opaco, com menor destaque do que o normal.

HTML:

```
<ul class="list-group">  
  <li class="list-group-item active">Item 1</li>  
  <li class="list-group-item disabled">Item 2</li>  
  <li class="list-group-item">Item 3</li>  
</ul>
```

NAVEGADOR:

Item 1
Item 2
Item 3

Exemplo de

lista com links. Aqui, ao invés de e , usamos <div> e a própria tag <a> para os itens da lista.

List-group-item-action → Diz para o Bootstrap, de forma visual, que este item executará uma ação.

HTML:

```
<div class="list-group">  
  <a class="list-group-item list-group-item-action" href="#">Link 1</a>  
  <a class="list-group-item" href="#">Link 2</a>  
  <a class="list-group-item" href="#">Link 2</a>  
</div>
```


NAVEGADOR:

Link 1
Link 2
Link 2

list-group-flush → Remove algumas configurações visuais como bordas, mas os itens não perdem a característica de lista, trazendo um visual mais discreto.

HTML:

```
<div class="list-group list-group-flush">  
  <a class="list-group-item" href="#">Link 1</a>  
  <a class="list-group-item" href="#">Link 2</a>  
  <a class="list-group-item" href="#">Link 2</a>  
</div>
```

NAVEGADOR:

Link 1
Link 2
Link 2

Assim, como vimos na aula 12 sobre os alertas, podemos utilizar todas aquelas variações para destacar itens da lista, como por exemplo o comando **list-group-item-primary**.

HTML:

```
<ul class="list-group">  
  <li class="list-group-item">Item 1</li>  
  <li class="list-group-item list-group-item-primary">Item 2</li>  
  <li class="list-group-item list-group-item-danger">Item 3</li>  
</ul>
```

NAVEGADOR:

Item 1
Item 2
Item 3

badge - traz uma ideia visual de notificações.

Assim, como vimos na aula 12 sobre os alertas, podemos utilizar todas aquelas variações para destacar itens da lista, como por exemplo o comando **badge-primary**

badge-pill → 'pill(comprido)' traz um estilo alongado.'

HTML:

```

<ul class="list-group">
  <li class="list-group-item">
    Curso de PHP
    <span class="badge badge-primary badge-pill">12 aulas</span>
  </li>
  <li class="list-group-item">
    Curso de javascript
    <span class="badge badge-primary badge-pill">15 aulas</span>
  </li>
  <li class="list-group-item">
    HTML
    <span class="badge badge-primary badge-pill">20 aulas</span>
  </li>
</ul>

```

NAVEGADOR:

Curso de PHP	12 aulas
Curso de javascript	15 aulas
HTML	20 aulas

d-flex → Classe do Bootstrap que ativa o display flex

justify-content-between → Classe já vista na aula 05 sobre tabelas

HTML:

```

<ul class="list-group">
  <li class="list-group-item d-flex justify-content-between align-items-center">
    Curso de PHP
    <span class="badge badge-primary badge-pill">12 aulas</span>
  </li>
  <li class="list-group-item d-flex justify-content-between align-items-center">
    Curso de javascript
    <span class="badge badge-primary badge-pill">15 aulas</span>
  </li>
  <li class="list-group-item d-flex justify-content-between align-items-center">
    HTML
    <span class="badge badge-primary badge-pill">20 aulas</span>
  </li>
</ul>

```

Curso de PHP	12 aulas
Curso de javascript	15 aulas
HTML	20 aulas

Também podemos utilizar a estrutura abaixo para criar um estilo de notícias:

HTML:

```
<div class="list-group">
  <a class="list-group-item list-group-item-action" href="#">
    <div class="d-flex justify-content-between">
      <h5>Título da Notícia</h5>
      <small>2 dias atrás</small>
    </div>
    <p>
      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Hic atque magnam eum autem accusantium fuga cumque dolores inventore explicabo? Facere explicabo ut corrupti, quia fugiat dicta. Facilis asperiores perferendis deleniti.
    </p>
  </a>
  <a class="list-group-item list-group-item-action" href="#">
    <div class="d-flex justify-content-between">
      <h5>Título da Notícia</h5>
      <small>2 dias atrás</small>
    </div>
    <p>
      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Hic atque magnam eum autem accusantium fuga cumque dolores inventore explicabo? Facere explicabo ut corrupti, quia fugiat dicta. Facilis asperiores perferendis deleniti.
    </p>
  </a>
  <a class="list-group-item list-group-item-action" href="#">
    <div class="d-flex justify-content-between">
      <h5>Título da Notícia</h5>
      <small>2 dias atrás</small>
    </div>
    <p>
      Lorem ipsum, dolor sit amet consectetur adipisicing elit. Hic atque magnam eum autem accusantium fuga cumque dolores inventore explicabo? Facere explicabo ut corrupti, quia fugiat dicta. Facilis asperiores perferendis deleniti.
    </p>
  </a>
</div>
```

NAVEGADOR:

<div>Título da Notícia</div> <div>2 dias atrás</div> <div>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Hic atque magnam eum autem accusantium fuga cumque dolores inventore explicabo? Facere explicabo ut corrupti, quia fugiat dicta. Facilis asperiores perferendis deleniti.</div>
<div>Título da Notícia</div> <div>2 dias atrás</div> <div>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Hic atque magnam eum autem accusantium fuga cumque dolores inventore explicabo? Facere explicabo ut corrupti, quia fugiat dicta. Facilis asperiores perferendis deleniti.</div>
<div>Título da Notícia</div> <div>2 dias atrás</div> <div>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Hic atque magnam eum autem accusantium fuga cumque dolores inventore explicabo? Facere explicabo ut corrupti, quia fugiat dicta. Facilis asperiores perferendis deleniti.</div>

Aula 21 – modal

Definição: Modal → Janela fake que aparece sobre o site. Quando o site escurece e aparece uma janela sobre o site(modal)

Aqui, para praticarmos a funcionalidade do modal, criamos um button e adicionamos as seguintes propriedades:

- **class="btn btn-primary"** - Para estilização do botão (já estudado)
- **data-toggle="modal"** - Também já visto. O toggle trás a funcionalidade de fazer e desfazer algo em um item.
- **data-target="#janela"** - Que é o alvo (target) da nossa ação.

```
<button class="btn btn-primary" data-toggle="modal" data-target="#janela">Clique aqui para abrir</button>
```

Para criar um modal, precisamos seguir a seguinte estrutura básica:

modal → Classe Bootstrap para a criação do modal.

fade → Classe que proporciona a animação (efeito) de ir aparecendo aos poucos (conteúdo já visto anteriormente).

id="janela" → Id que vincula o modal com o nosso button.

```
<div class="modal fade" id="janela">
```

modal-dialog → Cria a janela do modal.

modal-dialog-centered → Esta classe centraliza a janela do modal na tela.

modal-lg → Faz com que a largura do modal seja maior do que o padrão.

modal-sm → Faz com que a largura do modal seja menor do que o padrão.

```
<div class="modal-dialog modal-dialog-centered modal-lg">
```

modal-content → Todo o conteúdo do nosso modal deverá ficar dentro (sob) esta classe.

```
<div class="modal-content">
```

O modal oferece três áreas que PODEM ser usadas, de acordo com a necessidade:

- Cabeçalho: modal-header
- Corpo: modal-body
- Rodapé: modal-footer

Para a nossa aula utilizaremos as três para que tenhamos o conhecimento de todas elas.

modal-header → Cabeçalho do modal.

```
<div class="modal-header">
```

modal-title -Indica o título do modal.

```
<h5 class="modal-title">Título do Modal</h5>
```

data-dismiss="modal" → Este componente que realiza o fechamento do modal.

`×` -Adiciona um 'X', neste caso, no header

```
<button class="close" data-dismiss="modal"><span>&times;</span></button>
```

modal-body → Conteúdo (corpo) do modal.

```
<div class="modal-body">
```

modal-footer → Rodapé do modal.

```
<div class="modal-footer">
```

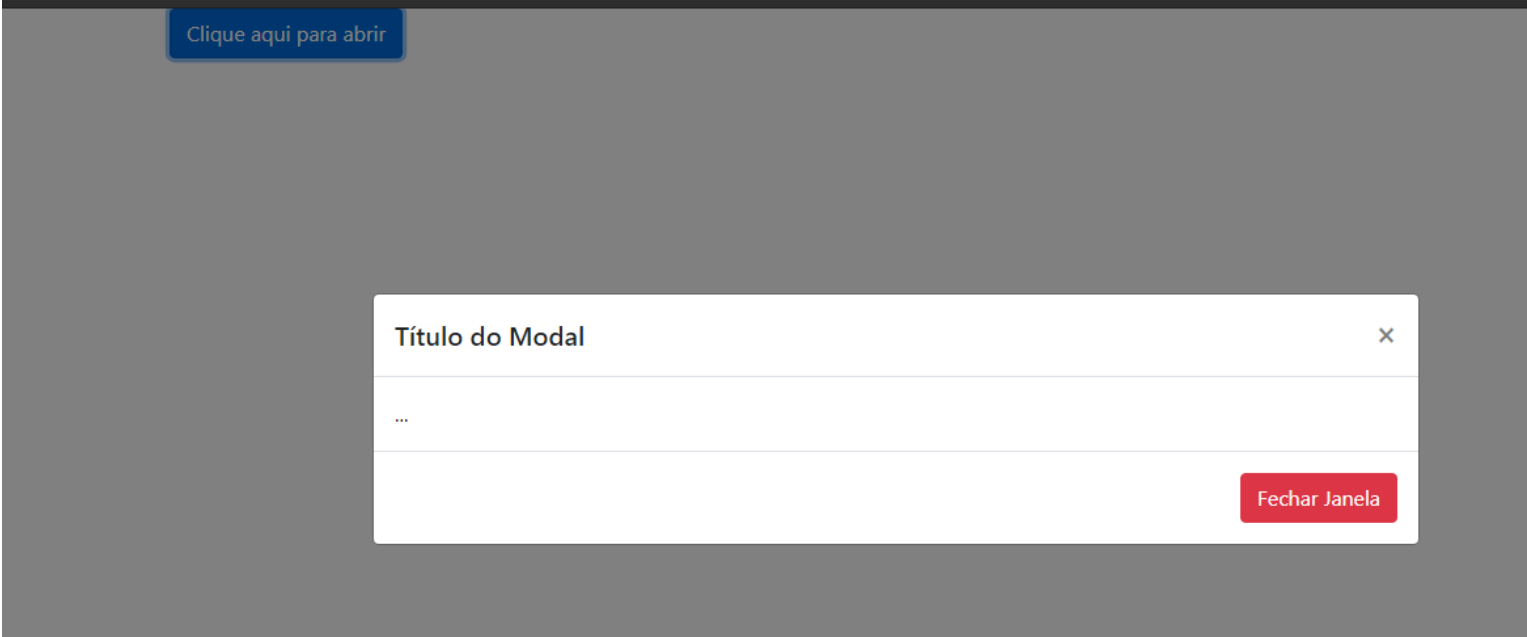
`<button class="btn btn-danger" data-dismiss="modal">Fechar Janela</button>` → Outro modelo de botão de fechamento.

```
<button class="btn btn-danger" data-dismiss="modal">Fechar Janela</button>
```

HTML COMPLETO:

```
<div class="container">
  <button class="btn btn-primary" data-toggle="modal" data-target="#janela">Clique aqui para abrir</button>
  <div class="modal fade" id="janela">
    <div class="modal-dialog modal-dialog-centered modal-lg">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title">Título do Modal</h5>
          <button class="close" data-dismiss="modal"><span>&times;</span></button>
        </div>
        <div class="modal-body">
          ...
        </div>
        <div class="modal-footer">
          <button class="btn btn-danger" data-dismiss="modal">Fechar Janela</button>
        </div>
      </div>
    </div>
  </div>
</div>
```

RESULTADO NO NAVEGADOR:



Aula 22 – navbar (Menu de Navegação)

navbar → Classe Bootstrap para estilização de barra de navegação.

navbar-expand-lg → Esta classe auxilia na responsividade em telas menores, criando uma indicação com 3 barrinhas para informar que há mais itens do que estão sendo mostrados.

navbar-dark → Esta classe é opcional e é utilizada apenas para caráter de estilização.

br-dark e **bg-light** → Também é opcional, ela adiciona um background(bg) à navbar. Neste caso, para a configuração de background, podemos utilizar também os recursos padrões do CSS.

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
```

navbar-brand → Área específica e **opcional** para adicionar nome, logo ou qualquer outro tipo de conteúdo. Estiliza como um título, ou seja, com um destaque sobre o restante do conteúdo.

```
<a href="#" class="navbar-brand">Empresa XYZ</a>
```

navbar-toggler → Trás a função de fazer e desfazer para o nosso botão

data-toggle="collapse" → Adiciona a função de collapse a ele.

data-target="#navbarMenu" → Vincula o clique do botão a nossa navbar-collapse.

```
<button class="navbar-toggler" data-toggle="collapse" data-target="#navbarMenu">
```

navbar-toggler-icon → Classe que cria aquele botão com 3 barrinhas.

```
<span class="navbar-toggler-icon"></span>
```

navbar-collapse collapse - Caso, o conteúdo não caiba na tela, utilizamos estas duas classes para que nossa navbar se adapte ao tamanho de tela utilizada.

id="navbarMenu" → Criado este id, para vincularmos esta função de collapse a um botão que aparecerá apenas em telas pequenas(celular).

```
<div class="navbar-collapse collapse" id="navbarMenu">
```

navbar-nav → Classe que inicia a declaração do nosso menu da navbar.

```
<div class="navbar-nav">
```

nav-item → Declaração de um item da navbar.

nav-link → Especifica este item como link.

```
<a href="#" class="nav-item nav-link active">Link1</a>
```

Ainda podemos adicionar mais conteúdo no restante da área livre da nossa navbar, fora da nossa navbar collapse. Conforme o exemplo abaixo:

```
<form method="POST" class="form-inline">
```

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

```
</form>
```

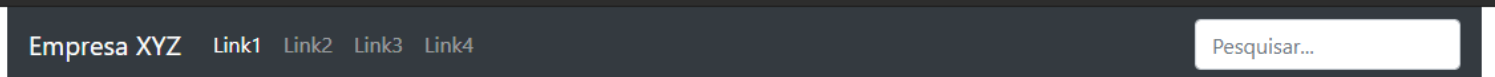
HTML COMPLETO:

```

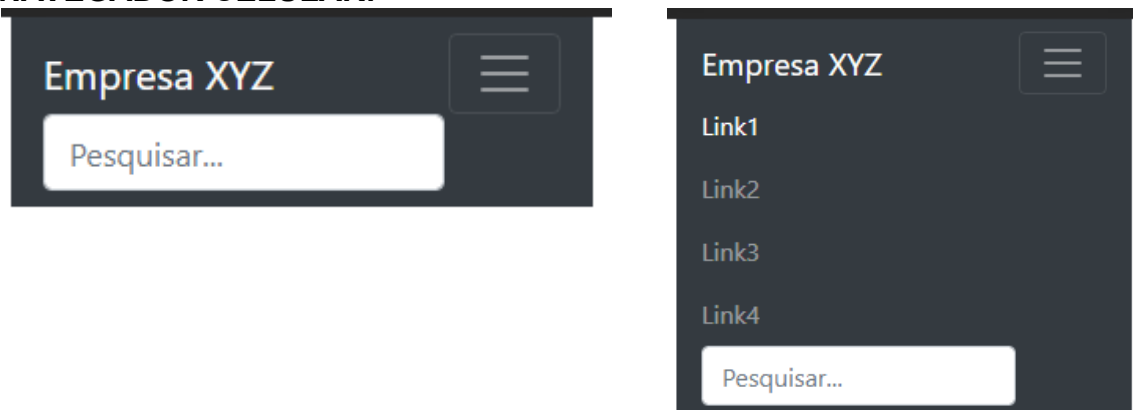
<div class="container">
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <a href="#" class="navbar-brand">Empresa XYZ</a>
    <button class="navbar-toggler" data-toggle="collapse" data-target="#navbarMenu">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="navbar-collapse collapse" id="navbarMenu">
      <div class="navbar-nav">
        <a href="#" class="nav-item nav-link active">Link1</a>
        <a href="#" class="nav-item nav-link">Link2</a>
        <a href="#" class="nav-item nav-link">Link3</a>
        <a href="#" class="nav-item nav-link">Link4</a>
      </div>
    </div>
    <!--Ainda podemos adicionar mais conteúdo no restante da área livre da nossa navbar, fora da nossa
    navbar collapse. Conforme o exemplo abaixo:-->
    <form method="POST" class="form-inline">
      <input type="text" class="form-control" placeholder="Pesquisar...">
    </form>
  </nav>
</div>

```

NAVEGADOR:



NAVEGADOR CELULAR:



Aula 23 – Paginação

Definição: Paginação são aqueles números que aparecem, geralmente na parte inferior de uma página para indicar o número de páginas que existem com um determinado conteúdo. Exemplo, o Google quando realizarmos alguma pesquisa e ele mostra lá embaixo a quantidade de páginas que contém o resultado da pesquisa. O Bootstrap já trás pra nós uma estilização para esta função de paginação.

pagination → Classe Bootstrap para estilização de paginação

pagination-lg → Classe que estiliza o tamanho dos botões lg(grandes). Também podemos utilizar a classe **pagination-sm** para estilizarmos os botões com tamanhos menores do que o padrão

justify-content-center → Classe Bootstrap já vista em outras aula que funciona dentro de uma `class="container"`.

```
<ul class="pagination pagination-lg justify-content-center">
```

page-item → Classe que define os itens desta paginação.

page-link → Classe que define os links desta paginação.

```
<li class="page-item active"><a href="#" class="page-link">1</a></li>
```

HTML COMPLETO:

```
<div class="container">
  <ul class="pagination pagination-lg justify-content-center">
    <li class="page-item disabled"><a href="#" class="page-link">Anterior</a></li>
    <li class="page-item active"><a href="#" class="page-link">1</a></li>
    <li class="page-item"><a href="#" class="page-link">2</a></li>
    <li class="page-item"><a href="#" class="page-link">3</a></li>
    <li class="page-item"><a href="#" class="page-link">4</a></li>
    <li class="page-item"><a href="#" class="page-link">5</a></li>
    <li class="page-item"><a href="#" class="page-link">Próximo</a></li>
  </ul>
</div>
```

NAVEGADOR:

Anterior	1	2	3	4	5	Próximo
----------	---	---	---	---	---	---------

Aula 24 – tooltip (ferramenta dica)

data-toggle="tooltip" → Marca o elemento para receber a funcionalidade tooltip

title="Ao clicar, irá salvar os dados" → Definimos a mensagem que irá aparecer

data-placement="right" → Define qual lado do botão irá aparecer a mensagem.

```
<button class="btn btn-success" data-toggle="tooltip" data-placement="right" title="Ao clicar, irá salvar os dados">Salvar</button>
```

```
<script type="text/javascript">
  $(function(){
    $('[data-toggle="tooltip"]').tooltip()
  })
</script>
```

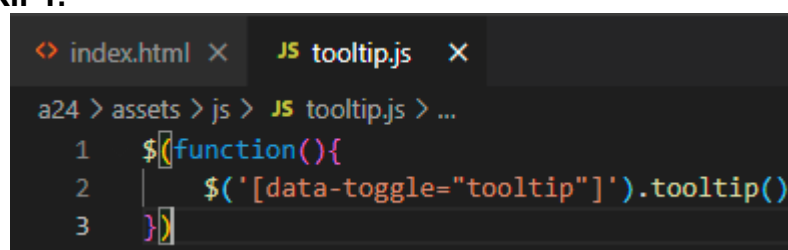
Para que o tooltip seja ativado, é necessário incluir no documento, abaixo dos scripts do jquery e do bootstrap, o seguinte script acima. Esse **script** procura no documento **HTML** todo o elemento que possui **data-toggle="tooltip"** e ativa o tooltip neles. Realizei o teste deste script no documento script.js e ele, primeiramente não funcionou. Para que ele funcionasse, tive que colocar o script.js após os documentos javascript do jquery e do bootstrap, conforme exemplo abaixo.

SCRIPT DENTRO DO DOCUMENTO HTML:

```
</div>
<script type="text/javascript" src="assets/js/script.js"></script>
<script type="text/javascript" src="assets/js/jquery-3.6.0.min.js"></script>
<script type="text/javascript" src="assets/js/bootstrap.bundle.min.js"></script>
<script type="text/javascript">
  $(function(){
    $('[data-toggle="tooltip"]').tooltip()
  })
</script>
</body>
</html>
```

MOVENDO ESTE SCRIPT PARA DENTRO DE UM DOCUMENTO DO JAVASCRIPT, É NECESSÁRIO ATIVAR ESTE SCRIPT APÓS O SCRIPT DO JQUERY E DO BOOTSTRAP. VEJA NO EXEMPLO ABAIXO EM QUE MOVI O SCRIPT PARA DENTRO DE UM DOCUMENTO QUE CHAMEI DE tooltip.js:

DOCUMENTO JAVASCRIPT:



```
index.html x JS tooltip.js x
a24 > assets > js > JS tooltip.js > ...
1  $(function(){
2    $('[data-toggle="tooltip"]').tooltip()
3  })
```

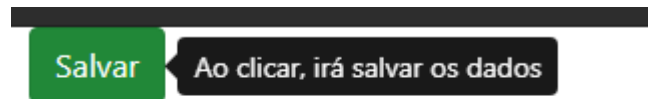
HTML:

```
</div>
<script type="text/javascript" src="assets/js/script.js"></script>
<script type="text/javascript" src="assets/js/jquery-3.6.0.min.js"></script>
<script type="text/javascript" src="assets/js/bootstrap.bundle.min.js"></script>
<script type="text/javascript" src="assets/js/tooltip.js"></script>
</body>
</html>
```

HTML COMPLETO:

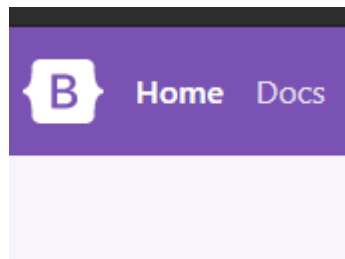
```
<div class="container">
  <button class="btn btn-success" data-toggle="tooltip" data-placement="right" title="Ao clicar, irá salvar os dados">Salvar</button>
</div>
<script type="text/javascript" src="assets/js/script.js"></script>
<script type="text/javascript" src="assets/js/jquery-3.6.0.min.js"></script>
<script type="text/javascript" src="assets/js/bootstrap.bundle.min.js"></script>
<script type="text/javascript">
  $(function(){
    $('[data-toggle="tooltip"]').tooltip()
  })
</script>
```

NAVEGADOR:



Aula 25 – Utilidades

<https://getbootstrap.com/>



Docs

Menu à esquerda:

▼ Getting started

- Introduction
- Download
- Contents
- Browsers & devices
- JavaScript
- Build tools
- Webpack
- Parcel
- Accessibility
- RFS
- RTL

> Customize

> Layout

> Content

> Forms

> Components

> Helpers

> Utilities

> Extend

> About

Migration

A dica aqui é ir em Utilities:

