**Tamanho de telas: class text-<tela>-<alinhamento>**

* Small >= 576px – sm
* Medium >= 768px – md
* Large >= 992px – lg
* Extra Large >= 1200px – xg

**Cores/Classes contextuais**primary, secondary, info, success (verde), warning (amarelo), danger (vermelho), light (“branco”), dark (“preto”), white (branco), black (preto)

Ex: bg-<cor>, text-<cor>

Opacidade = text-<cor>-<op%>

**Conteúdo principal**

Classes: container (centraliza), container-fluid (ocupa 100%)

**FORMATAÇÃO DE TEXTOS**

**Header (h1, h2, h3, h4, h5, h6)**  
Classes = h1, h2, h3, h4, h5, h6 (não só em headers)

**Display (maior que header)**  
Classes = display-1, display-2, display-3, display-4

**Parágrafos (p) (não só para p)**  
Classes = lead (destaque maior), text-monospace (monoespaçada)

Classes de estilo = font-weight-bold (negrito), font-weight-normal, font-italic

Classes de transformação de textos = text-uppercase, text-capitalize (primeiras letras maiúsculas), text-lowercase

Classes de alinhamentos = text-right, text-left, text-center

Classes de citação = blockquote (bloco de citação, como a tag), blockquote-footer (rodapé da citação, com footer e cite)

Classes de truncar/trancar = text-truncate (cortar frase de acordo com o tamanho da tela)

**Listas**Classes = list-unstyled (sem marcador), list-inline > list-inline-item (lista inline)

**Background**bg-<cor>  
bg-transparent (fundo transparente)

**ALINHAMENTO DE TEXTOS**

Parágrafo justificado = text-justify

**Display de alinhamento**Classes = d-inline, d-block, d-inline-block

**Float**

* float-left
* float-right
* float-none
* float-<tela>-<left / right> (responsivo)  
  No elemento pai dos float, podemos usar a classe clearfix para consertar a sobreposição dos elementos

**Conteúdo Fixo**  
fixed-<top / bottom>

**Colar no topo**  
sticky-top (conteúdo fixa no topo quando chega no topo ou fundo)

**Margin**

* m-<valor>
* Valores: 0 a 5
* mt (margin top)
* mb (margin bottom)
* ml (margin left)
* mr (margin right)
* mx (margin horizontal)
* my (margin vertical)
* m (margin)

**Padding**

* p-<valor>
* Valores: 0 a 5
* pt (padding top)
* pb (padding bottom)
* pl (padding left)
* pr (padding right)
* px (padding horizontal)
* py (padding vertical)
* p (padding)

**TAMANHOS E BORDAS**

**Largura/Width** w-<valor% / auto>

**Altura/Height**  
 h-<valor% / auto>

**Borda/Border**

* border
* border-top
* border-bottom
* border-right
* border-left
* border-<cor> (apenas para escolher a cor)

**Radius:**

* rounded (arredondada)
* rounded-top
* rounded-bottom
* rounded-right
* rounded-left
* rounded-circle (círculo/elipse)

**MEDIA QUERIES/MEDIA QUERY**

Tipos de mídias/dispositivos (para layout responsivo):

* all – todos os dispositivos
* aural – sintetizadores de voz
* braille – leitores de Braille
* embossed – impressoras de Braille
* handheld – dispositivos de mão. Por exemplo: celulares com telas pequenas.
* print – impressoras convencionais
* projection – apresentações de slides
* screen – monitores coloridos
* tty – teleimpressores e terminais
* tv – televisores

Exemplo de utilização:  
<link rel="stylesheet" media="print" href="print.css" />

@media (condição) {formatação}  
Ex: @media (min-width: 576px) and (max-width: 767.98px) {  
 body {  
 background: red;  
 }  
 }

**BOTÕES**

* Serve para <button>, <a>, <input type=”button”>, type=”submit”, type=”reset”, etc
* btn (padrão, usado com as outras)
* btn-<cor>
* btn-link (para parecer um link)

**Contorno:**

* btn-outline-<cor>

**Tamanho:**

* btn-<sm / lg>
* btn-block (bloco de botão)

**Estados:**

* active
* disabled
* data-toggle=”button” (alternar entre normal e ativo)

**Grupos:**

* Horizontal: btn-group (dentro da div que tem os botões)
* Vertical: btn-group-vertical (dentro da div que tem os botões)
* Toolbar: btn-toolbar (retira espaçamentos, ex: div toolbar > div group + div group
* Dropdown: dropdown na div;  
  dropdown-toggle e data-toggle=”dropdown” no botão;  
  dropdown-menu na div com os links e dropdown-item nos links dentro da div par associar com o botão  
  dropdown-divider (divisor entre os links dentro do dropdown)

**BARRAS DE NAVEGAÇÃO**

* nav (na ul, navegação simples, limpa formatação e deixa inline)
* nav-item (no li)
* nav-link (no a dentro do li)

Tipos de navegação (depois do nav):

* nav-pills (botões, + colocar active no item ativo da página)
* nav-tabs (abas, + active na página ativa)

Alinhar navegação (na ul):

* justify-content-center
* justify-conten-end (à direita)
* flex-column (navegação vertical)

Barra de navegação simples:

* <nav>
* link de logo num link solto dentro de <nav>
  + classe navbar-brand
* dentro da <nav>, classe navbar
  + para deixar inline,+ classe navbar-expand-<tela>
  + cor: navbar-<dark/light>
  + background: bg-<cor>
* dentro da <ul>, classe navbar-nav
* nav-item dentro dos <li>

Barra de navegação com menu responsivo (menu hamburguer):

* <button> com classe navbar-toggler e atributo data-toggle=”colapse” e data-target=”#id da div”
* > <span> com classe navbar-toggler-icon
* <div> com <ul> dentro, classe collapse + navbar-collapse (para fazer sumir a navegação, a não ser que clique no hamburguer)

Barra de navegação com formulário (search bar/barra de pesquisa):

* Dentro da div com navegação simples, <form> com classe form-inline
* <input> type=”text” com classe form-control
* <button> de pesquisar

Barra de navegação com menu dropdown:

* Barra de navegação simples ou com formulário...
* Dentro do <li> que deseja criar o dropdown, classe dropdown e criar uma <div> com os links desejados
* Dentro do link do <li>, classe dropdown-toggle, data-toggle=”dropdown”
* Dentro da <div>, classe dropdown-menu
* Dentro dos sublinks na <div>, classe dropdown-item

Alinhamentos:

Em <nav>, classes fixed-top, fixed-bottom ou sticky-top

**LISTAS**

* Base: list-group (lista como se fosse tabela) > list-group-item
  + Funciona com links também
  + Pode receber active/disabled junto
* list-group + list-group-flush (com border top e bottom)
* Cores/Classes contextuais: list-group-item-<cor>
* Badges: list-group > list-group-item > <span> (dentro do <li>) com classe badge + badge-<cor>
* Breadcrumbs: para indicar em que página o usuário está
  + <ol> com classe breadcrumb
  + <li> com classe breadcrumb-item + nada ou active ou disabled

**FORMULÁRIOS**

* <form> > <div> (para cada item do formulário) com classe form-group (espaçamento entre itens)
* <input>/<select>/<textarea> com classe form-control + form-control-<tela>
* <small> com classe form-text para textinhos em baixo dos items
* Input de arquivos:
  + <input> com form-control-file para upload de arquivos
  + <div> com classe custom-file > <label> com classe custom-file-label + <input> com classe custom-file-input
* Range:
  + <label> com classe range
  + <input> com classe custom-range

Formulário inline:

* + <form> com classe form-inline
  + > <input>s com classe form-control

Validação de formulário (javascript):

* Classes: is-valid, is-invalid (borda verde ou vermelha), valid-feedback, invalid-feedbak (fica como a formatação do <small>)

**INPUT GROUPS**

Grupo prepend (conteúdo antes da caixa de texto):

* <div class=”input-group”> > <div class=”input-group-prepend”> > <span> ou <div> class=”input-group-text”>

Grupo append (conteúdo depois da caixa de texto):

* <div class=”input-group”> > <div class=”input-group-append”> > <span> ou <div> class=”input-group-text”>

Consegue fazer múltiplos inputs na mesma linha

**ALERTAS**

* Classes: alert + alert-<cor>
* Com botão de fechar:
  + alert + alert<cor> + alert-dismissible > button com classe close + data-dismiss=”alert”

**BARRA DE PROGRESSO**

* <div class=”progress”> > <div class=”progress-bar”> (com width para indicar o progresso)
* Cor: bg-<cor> (na div 2)
* Listrado: progress-bar-stripe (na div 2)
* Animada: progress-bar-animated (na div 2)
* Compatível com múltiplas barras na mesma div

**TABELAS**

* Dentro de <table>, classe table +:
  + table-<dark/light>
  + table-striped
  + table-bordered
  + table-borderless
  + table-<tamanho>
* <thead class=”<dark/light>”
* Cor/Classes contextuais: table-<cor> (em qualquer tag da table)
* Responsiva: dentro de uma <div>, table-responsive > <table> com table
* Paginação (separação por página):
  + <nav> >
  + <ul class=”pagination”> >
    - Alinhamento:
      * justify-content-<center/end>
    - Tamanho:
      * pagination-<sm/lg>
  + <li class=”page-item”> > (com active/disabled)
  + <a class=’’page-link”>
    - Setas:
      * <span> &laquo;/&raquo; </span> (entidades html)
      * Acessibilidade: <span class=”sr-only”>Anterior/Próximo</span> (screenreader)

**CARTÕES/CARDS**

Cartão simples:

* <div class=”card”> (bg, border, etc> <div class=”card-body”> > conteúdo (img, h, p, a, etc)
  + <hn class=”card-title”>
  + <hn class=”card-subtitle”>
  + <p class=”card-text”>
  + (fora do card-body) <img class=”card-img ou card-img-<top/bottom>

Cartão com imagem sobreposta:

* Primeira div: card-img-overlay

Cartão com cabeçalho:

* Nova div antes da primeira div: card-header

Cartão com cabeçalho e rodapé:

* Nova div depois da primeira div: card-footer

Cartão com navegação:

* <ul> dentro do card-header
* <ul>, <li>, <a> com classes de navegação
  + <ul class=”card-header-tabs”> (junto com as outras)
  + <a> com active/disabled

Cartão com colunas:

* <div class=”card-columns”> > divs de cada cartão

Alinhamento:

* Alinhamento de texto

**JUMBOTRON**

Área de destaque

* <div class=”jumbotron”> > conteúdo (classes de texto)

**ESTRUTURA PARA IMAGENS**

* <div class=”media”> > <img>, <div class=”media-body”> > conteúdo

Alinhamentos:

* Dentro de <img>: align-self-<start/center/end>

Listas com imagens:

* <ul> > <li class=”media”> > <img>, <div class=”media-body”> > conteúdo

**GRID**

Divisão da área de conteúdo em 12 colunas. A soma das colunas ocupadas tem que ser 12.

Linhas:

* .row (cria linha em flex container)

Colunas:

* Extra Small <576px, sem largura máxima do container
  + .col-<qtd de colunas>
* Small >=576px, largura máxima do container 540px
  + .col-sm-<qtd>
* Medium >=768px, largura máxima do container 720px
  + .col-md-<qtd>
* Large >=992px, largura máxima do container 960px
  + .col-lg-<qtd>
* Extra large >=1200px, largura máxima do container 1140px
  + .col-xl-<qtd>

Largura fixa:

Classe col deixa todas as divs com áreas iguais e sempre inline

Largura automática:

Caso não haja especificação da qtd de colunas, a div ocupará o que restou

Largura igual empilhada:

col-<tela>: daquela tela para baixo, as áreas serão empilhadas

Ordenação:

* order-<n inteiro>
* offset-<tela>-<qtd>: define a quantidade de colunas a pular a partir de uma tela (ex: centralizar)

Aninhamento:

* row dentro de row cria uma row que segue as regras/tamanho da primeira

Alinhamento:

Utiliza flex box:

* + Align-items<center/start/end> (horizontal)
  + Align-self-<center/start/end> (vertical)
  + Justify-content-<center/start/end> (horizontal das colunas, usado no row)

Distribuição:

Justify-content-<around/between>

Removendo espaçamento:

No-gutters

**FLEX CONTAINER**

Display-flex: d-flex +:

* Flex container e flex items:
  + Flex-row (padrão, muda para linha)
  + Flex-row-reverse
  + Flex-column (muda para coluna)
  + Flex-column-reverse
* Alinhamento horizontal:
  + Justify-content-<start/center/end/around/between>
* Colunas:
  + Flex-column
  + Flex-column-reverse
* Alinhamento vertical:
  + Align-items-<start/center/end/baseline/stretch>

Alinhamento self/individual (nos items):

* Align-self-<start/center/end/baseline/stretch>

Margin (nos itens):

* Mr/Ml-auto (margin right/left, centralizar horizontal)
* Mb/Mt-auto (margin bottom/top, centralizar vertical)

Wrap:

* Flex-nowrap (padrão, barra de rolagem)
* Flex-wrap (quebra para a próxima linha)

**ESCONDER CONTEÚDO**

d-none (display none) + d-<tela>-<block/inline/flex> (exibir apenas a partir daquela tela)