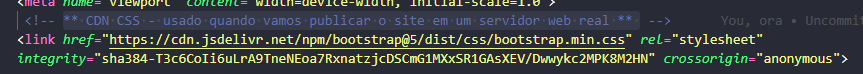
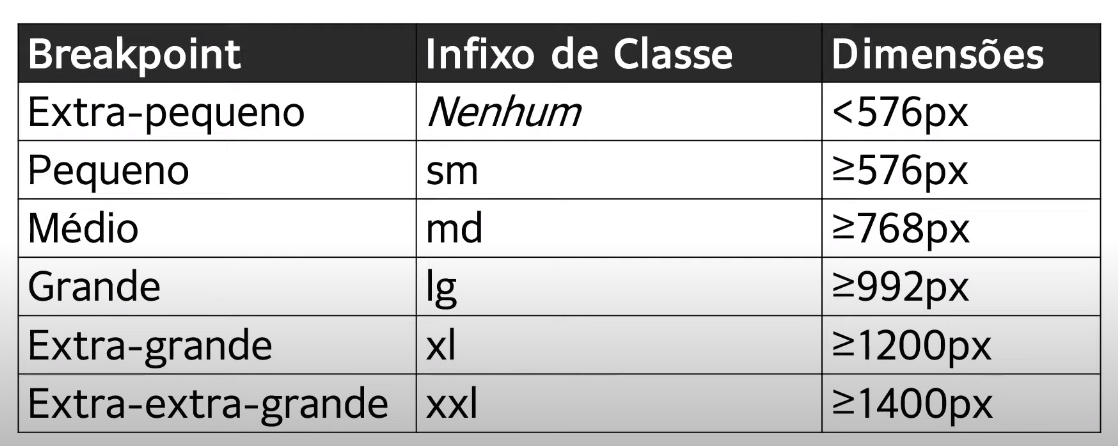
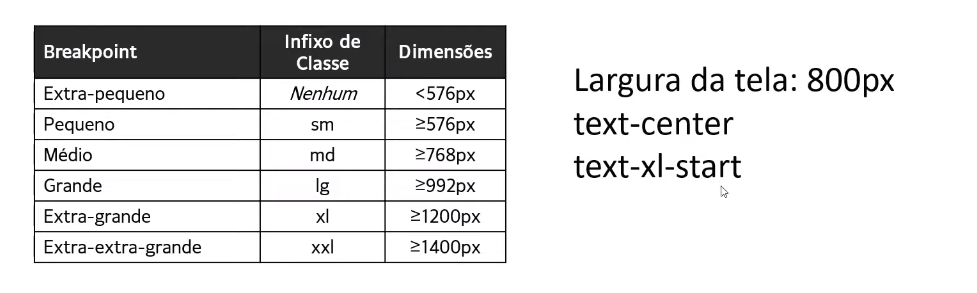
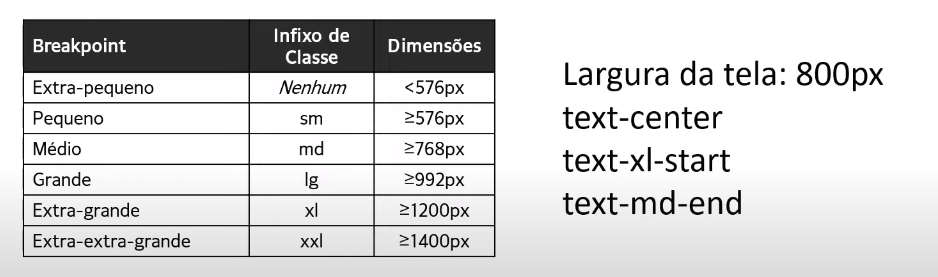
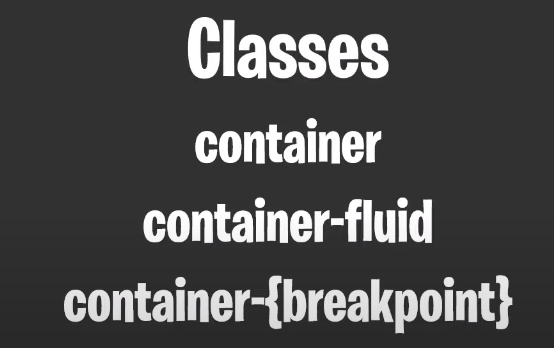
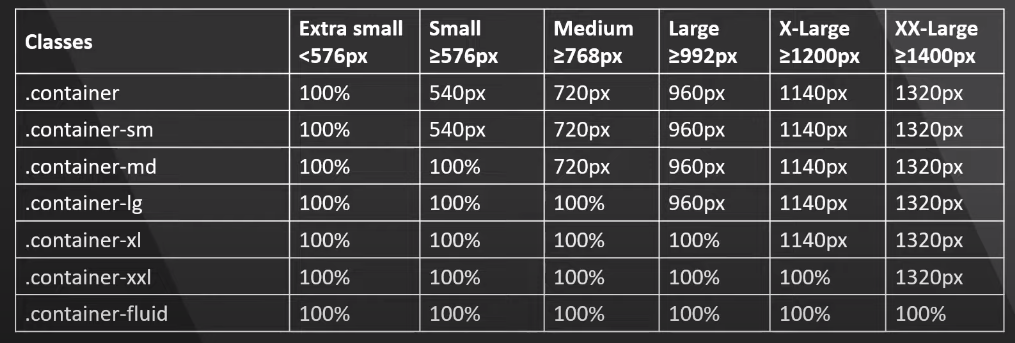
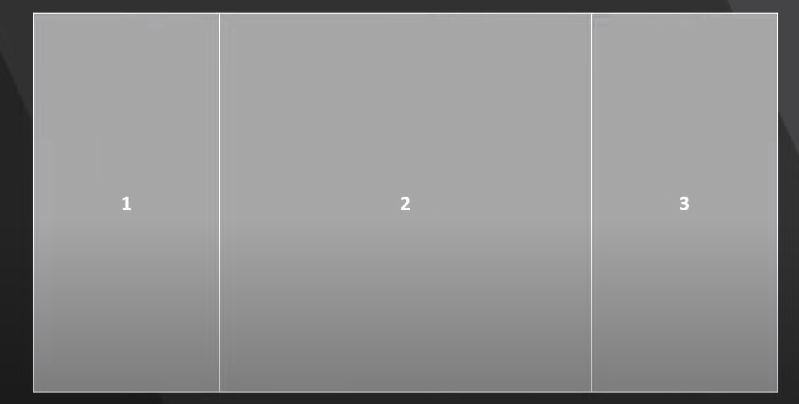
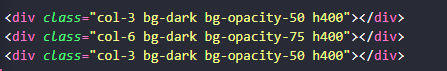
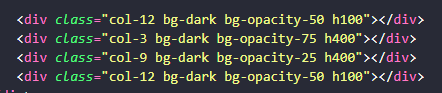
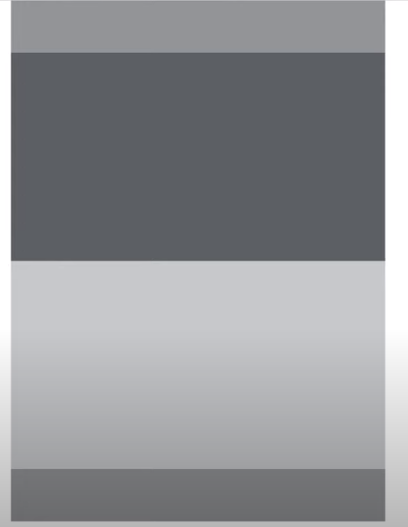
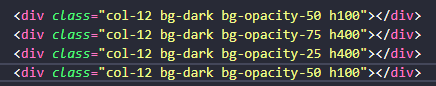
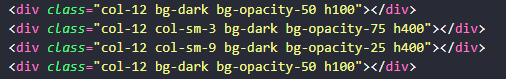
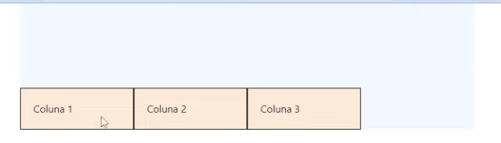
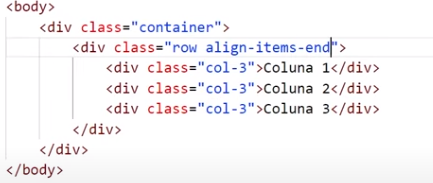
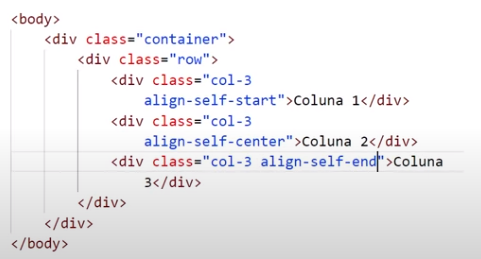
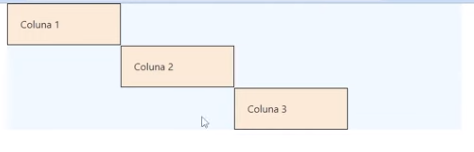
**Anotações – Aula Bootstrap 5   
Curso:** **https://www.youtube.com/playlist?list=PL0YuSuacUEWuJN3qb6NP15bzqd8w\_oAj7**

Ensina a instalar o bootstrap diretamente na pasta

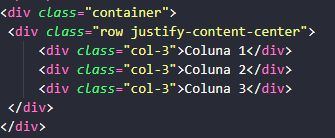
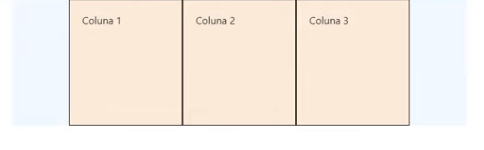
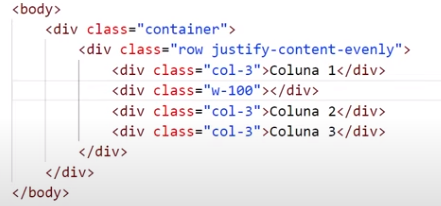
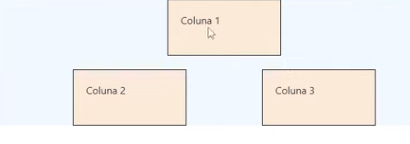
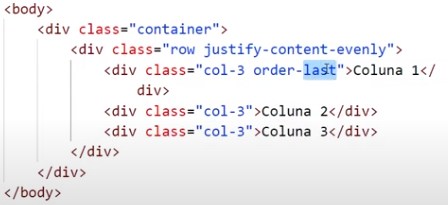
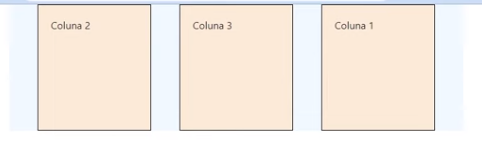
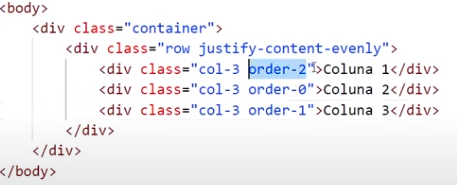
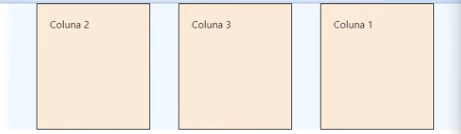
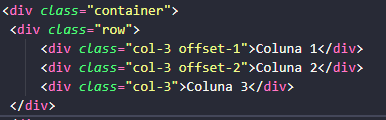
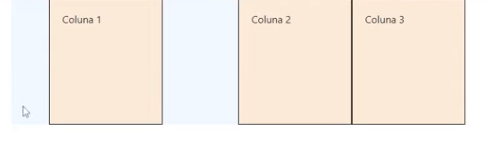
https://www.youtube.com/watch?v=Ka16scuMc0c&list=PL0YuSuacUEWuJN3qb6NP15bzqd8w\_oAj7&index=4    **Diferença CDN – Local  
  
CDN** - usado quando vamos publicar o site em um servidor web real   
  
Versão **LOCAL** DO NOSSO Bootstrap - usao quando o site está em fase de construção/desenvolvimento  
  
**Breakpoints – responsivo**   
https://getbootstrap.com/docs/5.0/layout/breakpoints/

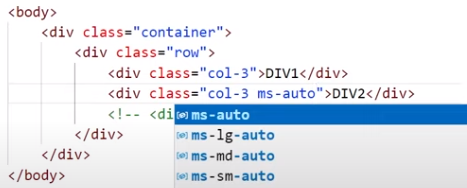
**Infixos de Classe:**   
  
**m-5** \*adiciona uma magin de 5 unidades de bootstrap em todo contorno do elemento que usa essa classe, não possui infixo, por isso é uma classe de breakpoints  
  
**p-sm-3** \*adiciona margens internas de 3 unidades a um elemento quando a resolução estiver no breakpoint pequena – desde que a classe possua o infixo sm, que vem small.  
  
**col-lg-2** \*define uma região com duas colunas de largura, quando a resolução de tela estiver na faixa breakpoints grande lg, que vem de large  
   
**fs-md-6** \*define um tamanho de fonte igual a 6 para um elemento, quando a resolução de tela estiver em médio md, que vem de medium  
  
**text-xl-end** \*alinha o texto de um elemento a direita, quando a resolução estiver grande xl, que vem de Extra Large.  
 *obs: quando a largura da tela é superior a todas as larguras de breakpoints CSS aplicadas a um elemtento, a classe com maior breakpoints é usada.*

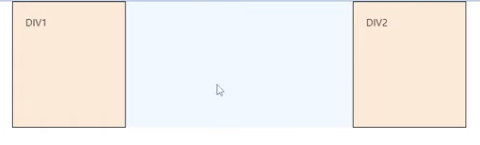
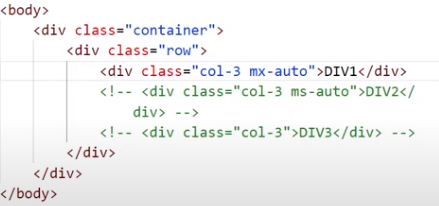
Utilizando esse exemplo: ****Neste exemplo o texto o ficará no inicio da nossa tela (start) quando a resolução estiver maior que 1200px.   
Então, para os breakpoints anteriores/inferiores aparecerá centralizado (center).  
  
  
Neste exemplo adicionamos o “text-md-end” que será para larguras maiores ou superiores a 768px, portanto o texto passaria a estar alinhado a direita nestas condições.  
  
**Contêineres:**  
bloco de construção  
  
  
**Classes**  
  
container:  
  
container-fluid: largura total (sempre ocupa 100% da largura disponivel)  
  
container-{breakpoint}: responsivo  
o “breakpoint” simboliza um dos infixos dessa classe.  
  
  
 **GRID:**divide em container – de iguais colunas  
  
****   
por padrão são divididas por 12 colunas (imaginárias)   
  
se criamos um layout com 3 colunas verticais, sendo que a 2 coluna cerá centralizada e as colunas 1 e 3 terão metade da largura da 2.   
Ou seja: Coluna 2 = 6 colunas / Coluna 1 e 3 = 3 colunas cada.  
Porém, as colunas são apenas limitadores virtuais de largura, imaginárias.   
  
  
E se desejamos que a coluna 3 tenha 9 linhas¿

****neste caso basta definir apenas col para a coluna 1 e 3:  
****  
Pois assim ela divide o restante dos espaços das colunas com as outras div's que tenham a class col  
  
Layout muito utilizado na web >  
  
  
  
definimos a primeira região com 12 colunas para ocupar a linha inteira, e após a 2 e 3 para ocupar a segunda linha, e a 4 com 12 colunas para poder ocupar também a linha inteira abaixo.  
  
**obs:** esse layout não é responsivo, para isso temos que acrescentar os breakpoints  
  
este é um exemplo de como deve aparecer o **layout mobile**  
com todas as col\*12  
  
  
Como o bootstrap foi desenvolvido para progetar primeiro em mobile para depois ir fazerndo a responsividade, vamos partir daqui para colocar nossos **breakpoints.**   
  
quando definimos col-sm-3 e col-sm-9, damos a responsividade necessária para nosso layout  
SM = 576px – ou seja, larguras maiores que 576px dicaram com o layout web:  
  
  
  
 **  
OBS:** quando colocamos “defer” ao final do nosso script JS, definimos que ele só irá carregar depois de todos os elementos da nossa página.  
Isso é o que vale colocar o nosso script no final do documento.  
  
  
**Formatação de Colunas de Layout**align-items-start  
align-items-center  
align-items-end **  
**quando queremos alinhar apenas nossas colunas, nós utilizamos, e esse alinhamento só é possível pois todo “row” é um container flexbox  
  
align-self-start  
align-self-center  
align-self-end  
  
**** 

alinhamento horizontal  
  
justify-content-start  
justify-content-center  
justify-content-end  
justify-content-between  
justify-content-around  
justify-content-evenly

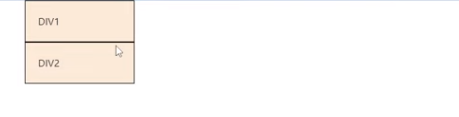
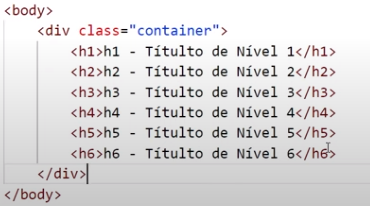
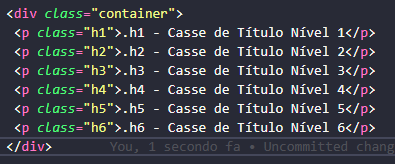
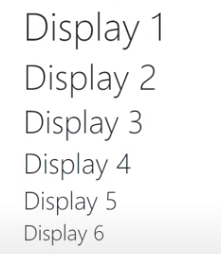
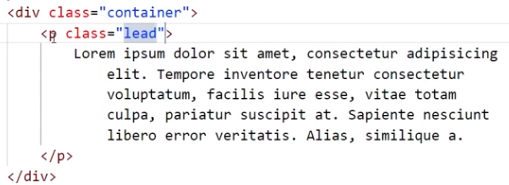
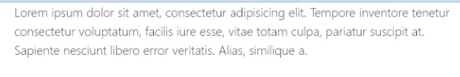
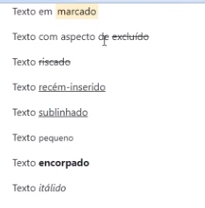
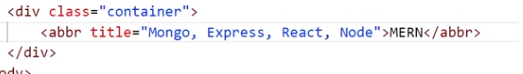
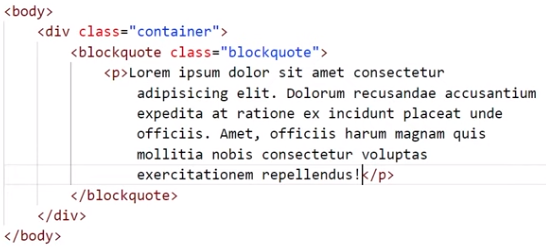
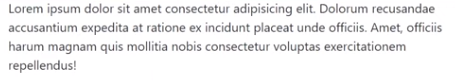
**  
**e se quisermos colocar a coluna 1 acima das colunas 2 e 3¿   
  
para isso devemos incluir uma div class=”w-100”  
  
****  
  
podemos também reordenar nossas colunas:   
  
order-last  
order-first  
  
  
  
  
Dessa forma a nossa coluna 1 ficará exposta na última posição  
  
  
podemos também incluir números na order  
  
****  
****  
  
Podemos também deslocar as nossas colunas:   
  
basta incluir "**offset-1**" na coluna 1, "**offset-2**" na coluna 2, e elas se deslocam a quantidade de posições à direita das 12 posições totais do container  
  
veja que ficou uma coluna deslocada antes da coluna 1, e duas na coluna 2  
  
****  
  
  
Podemos colocar margens nas colunas:   
  
podemos fazer a coluna 2 ir totalmente para a direita se dermos "**ms-auto**" para ela

dessa forma damos margin start (ms) para nossa coluna, e a margem será de todo o conteudo disponivel no container  


****vamos usar apenas a coluna 1 neste exemplo de margem:  
  
se colocarmos "**mx-auto**" (x vem do eixo horizontal) daremos margem para a esquerda e direita proporcionalmente  
****  
****

Definindo colunas soltas, sem um container de class=”row”

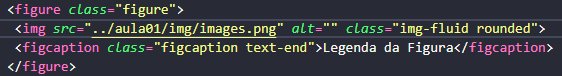
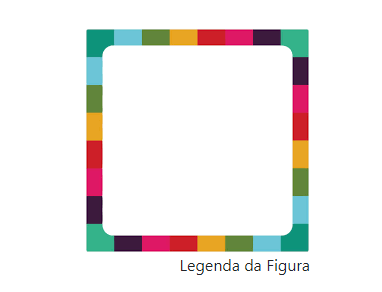
veja que apensar de compartilhar a mesma coluna, elas não ficam na mesma linha (uma ao lado da outra)  
****   
isso acontece porque estão fora de um container de class row

esse recurso é bom quando precisamos dimensionar div's  
****  
  
**Tipografia**por padrão o bootstrap já tem sua própria tipografia(font)  
****  
****  
  
Mas como podemos mudar nossa font/tipografia então¿   
  
**** + TAB =  
  
neste caso usamos paragrafos e definilos como h1, h2, h3, h4, h5 e h6  
  
Também podemos utilizar display  
****  
****   
dessa forma geramos 6 elementos do tipo H1, mas com diversos display (1 ao 6)  
****  
  
Textos de paragrafos comuns:  
  
 ****Usando p class="lead" - deixa o paragrafo melhor e uma fonte sofisticada, serve para destacar um trecho de texto ****  
  
Marcações de texto HTML:  
  
****  
  
**ABBR**  
  
Usando para abreviações – siglas e coisas do genero,   
  
****  
  
MERN é a abreviação de Mongo, Express, React, Node  
 - possui um sublinhado pontilhado   
  
Utilizando **Blockquote** :  
  
Bloco padrão de citação no bootstrap  
 **  
**

## Blockquote + Figure (figcation) Bloco padrão de citação no bootstrap + referencias autor + TAB = Formatação de lista: =

## list-unstyled = remove os marcadores da frente da lista e altera a margem Como colocar uma lista ao lado da outra: + TAB = utilizado para organizar itens que precisam estar em uma mesma linha (nav/menu) ul class="list-inline" + li class="list-inline-item" Lista de definição – Lista de descrições + TAB = \*dt = “{Termo}” \*dd= “{Descrição}” Imagens e Figuras transformando imagens em responsivas com Bootstrap, dessa forma elas não vão ultrapassar seu container além disso, deixa as imagens com um aspecto visual mais leve + TAB = img-fluid obedece o tamanho do container, portanto se nossa tela diminuir, logo a imagem também vai img-thumbnail dá bordas arredondada a nossa imagem rounded arredondadas a borda da nossa imagem

## img-thumbnail + float-start + p class="lead" + Tab = colocamos um lorem80 para preencer todo o P dessa forma o texto acompanhará a imagem Podemos dar uma margem de 3px na nossa imagem: \*me – margem à direita da imagem dessa forma o texto não ficará grudado na imagem para deixar nossa imagem no final do texto incluimos float-end e damos ms-3 (margem esqueda) anossa imagem

**Já para centralizar a imagem  
basta dar text-center ao nosso container**Exibição de figuras – com legenda  
  
 + TAB =   
  
**usamos figure + img-fluid rounded + figcaption** **Tabelas**Como configurar tabelas responsivas

por padrão o bootstrap não formata as tabelas por padrão quando você adiciona a referência dele dentro da página, é necessário que você aplique classes especificas para formatação de tabelas  
  
Nossa estrutura criada:

## + TAB = + TAB = -1 coluna com 4 linhas + TAB = - 5 colunas com 4 linhas cada No navegador ficará = tabela comum

## adicionando classe as tabelas "table" + cor "table-primary" = tabelas listrada (cor sim cor não) definimos na classe "table-striped" + no thead class="table-dark"

## definimos na classe "table-striped-columns" para deixar cada coluna com uma cor iluminando as tabelas/linhas definimos na classe "table-hover" para que ao passar o mouse elas fiquem levemente iluminadas para destaque \*não se vê no printscreen

## Podemos também incluir a class table-active para deixar uma linha selecionada em destaque

## Bordas nas tabelas definimos na classe table-bordered Retirando as bordas das tabelas definimos na classe table-borderless

## Tabelas compacta definimos na classe table-sm - faz com que o padding de todas as células caiam pela metade, gerando uma tabela mais compacta Tabelas com linha no tbody definimos na classe tbody class="table-group-divider" - que gera uma divisória mais forte entre o agrupamento tbody e o thead

## Tabela com conteudos grandes na coluna definimos na classe align-middle para alinhar os conteudos das células quando temos grandes conteudos que possam desconfigurar nossa célula

## dessa forma os conteúdos Coluna 2 e Coluna 3 ficaram alinhados com o texto Utilizando caption como titulo na Tabela definimos após o tbody um caption e se quisermos colocar o título acima da tabela definimos na classe da table caption-top

**Utilizando a barra de rolagem horizontal apenas na tabela**  
  
quando temos tabelas muito grandes, podemos definir para que apenas a tabela tenha uma barra de rolagem, e não a página toda  
  
para isso precisamos criar uma div class="table-responsive" antes da nossa table

## dessa forma apenas a tabela terá uma barra de rolagem, deixando responsivo

## Cores e Opacidade O bootstrap possui 8 cores temáticas Para colocarmos cores em nosso background basta incluir na class “bg-primary” possuimos também bg transparent e white além das 8 cores acima mencionadas, que são especificas para planos de fundo Colocando uma outra div e incluindo o fundo cinza conseguimos ver melhor na página web Gradient Podemos também incluir bg-gradient nas cores que puxa de uma cor mais fraca no topo do paragrafo, até uma cor escura no final Controlando a Opacidade do plano de fundo neste caso utilizamos bg-opacity + a porcentagem de opacidade que desejamos entre: 10, 25, 50, 75 e 100%. veja que no dark (preto), agora com a opacidade em 50% conseguimos ler seu conteudo. Cores da fonte com as cores das fontes podemos utilizar os mesmos infixos dos paragrafos. sendo que no paragrafo dark, incluimos a fonte white para que seja possível visualizar na tela. Temos tamtém a text-muted, que dá um aspecto cinza claro para o texto : para dar um destaque menor no texto. Opacidade da fonte podemos dar opacidade para as nossa fontes também, basta usar text-opacity, e incluir a porcentagem desejada:

## 

## Cores automáticas com destaque entre si podemos incluir text-bg + o infixo do nosso background = dessa forma, ele puxará a cor com mais contraste para o nosso background escolhido Alterando a cor de link + TAB = podemos alterar as cores dos links, que por padrão sempre é azul

## Mas podemos alterar no link + infixo da cor que desejamos:

## Icons (+ 1600)

## Paramos nessa aula!

## Botão de busca =

## 