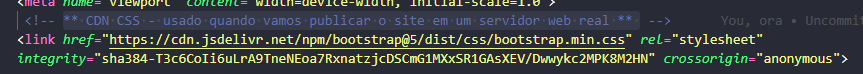
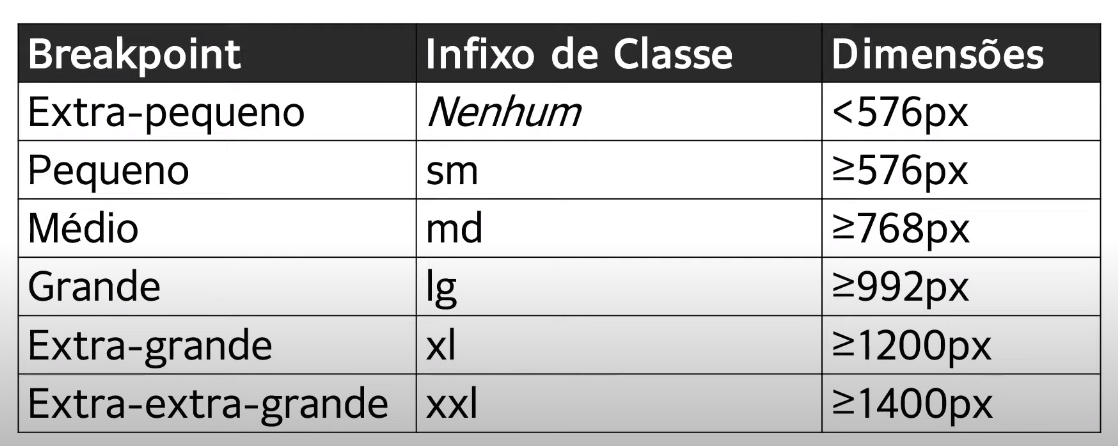
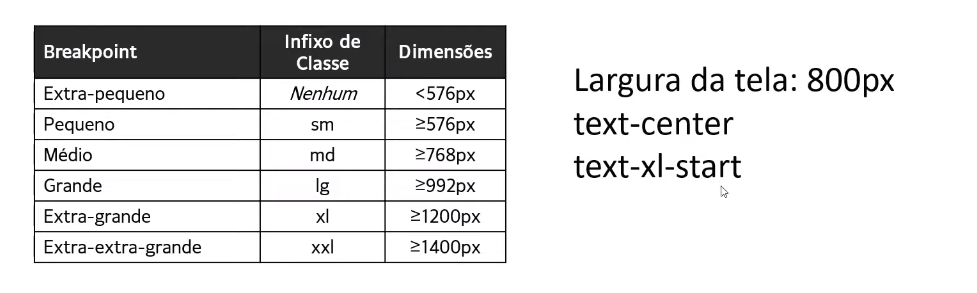
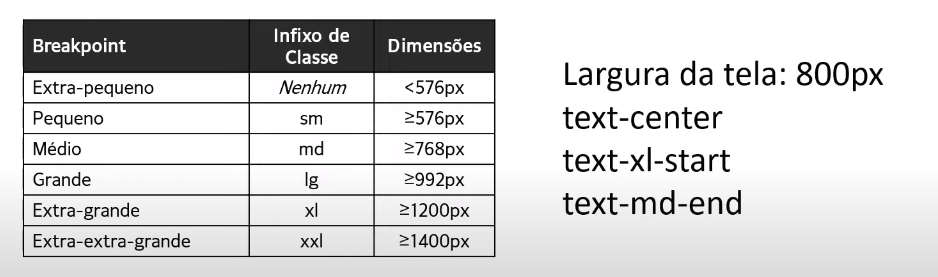
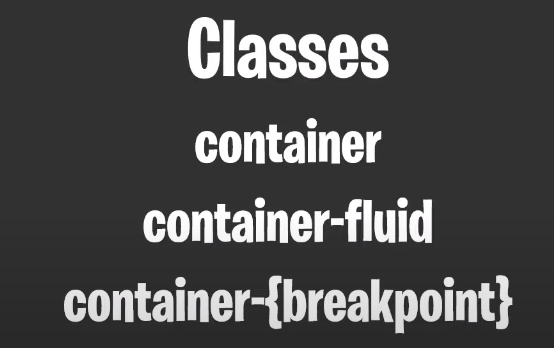
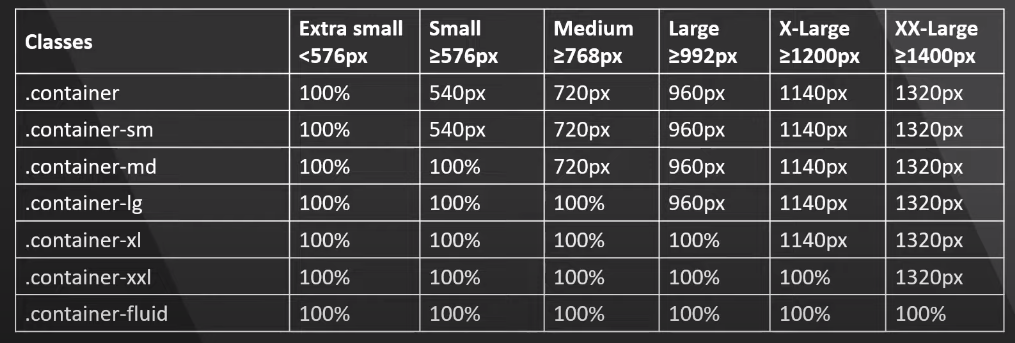
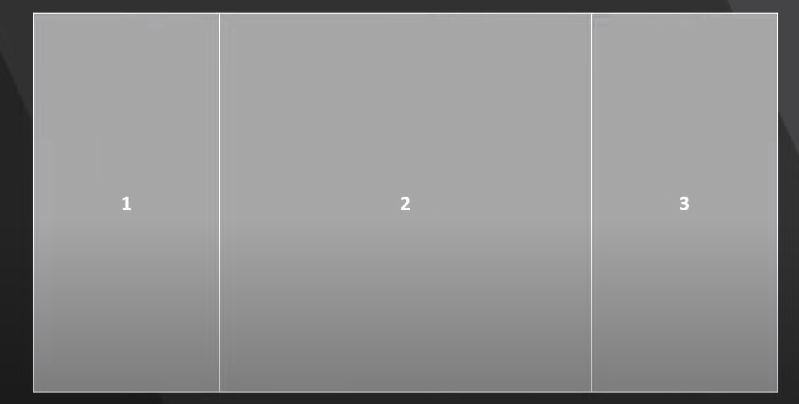
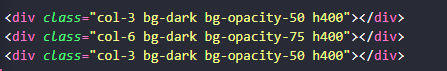
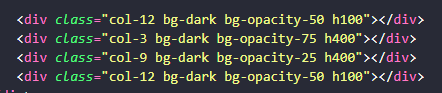
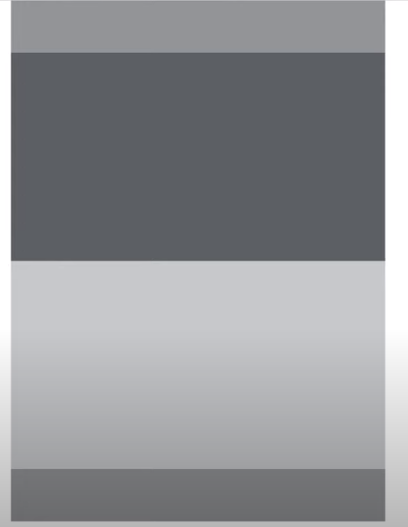
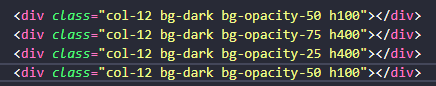
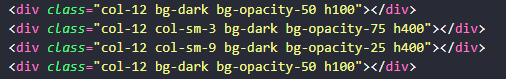
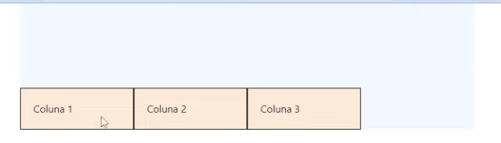
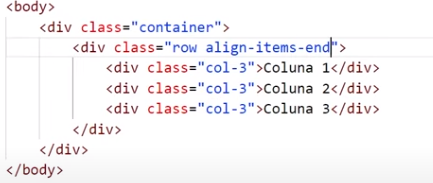
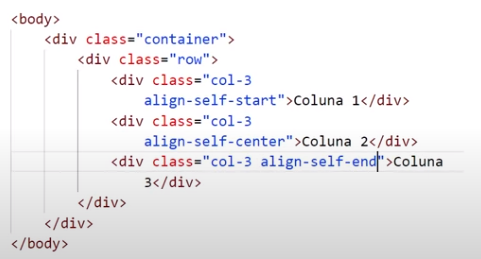
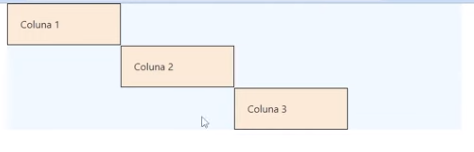
**Anotações – Aula Bootstrap 5**

Ensina a instalar o bootstrap diretamente na pasta

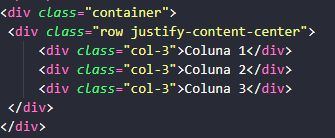
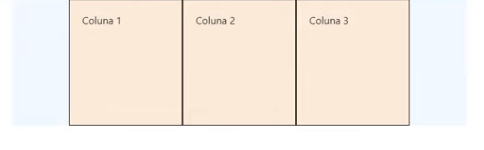
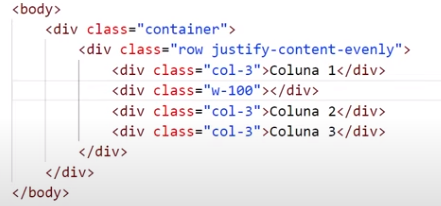
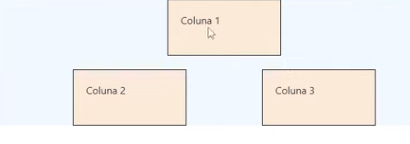
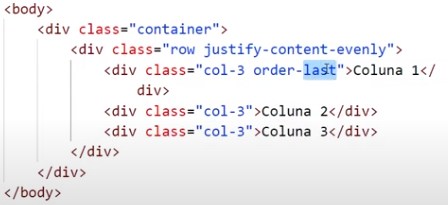
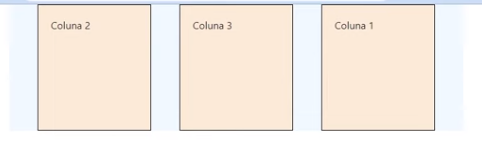
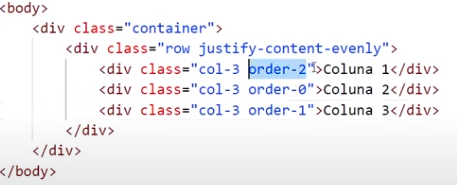
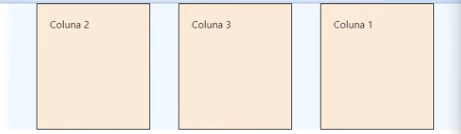
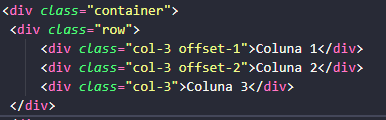
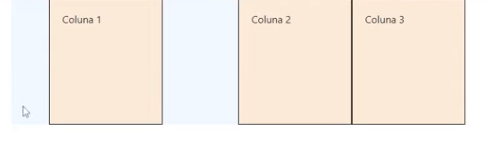
https://www.youtube.com/watch?v=Ka16scuMc0c&list=PL0YuSuacUEWuJN3qb6NP15bzqd8w\_oAj7&index=4    **Diferença CDN – Local  
  
CDN** - usado quando vamos publicar o site em um servidor web real   
  
Versão **LOCAL** DO NOSSO Bootstrap - usao quando o site está em fase de construção/desenvolvimento  
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

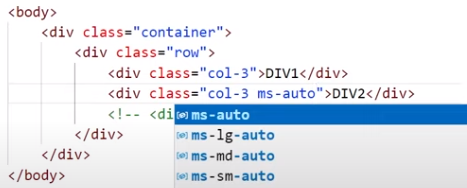
**Infixos de Classe:**   
  
**m-5** \*adiciona uma magin de 5 unidades de bootstrap em todo contorno do elemento que usa essa classe, não possui infixo, por isso é uma classe de breakpoints  
  
**p-sm-3** \*adiciona margens internas de 3 unidades a um elemento quando a resolução estiver no breakpoint 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.*

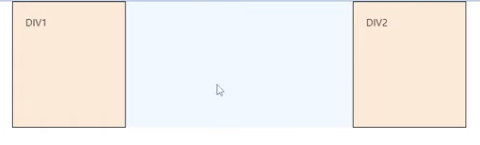
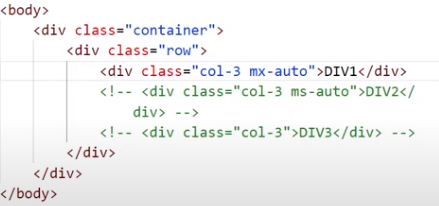
Utilizando esse exemplo: ****Neste exemplo o texto o ficará no inicio da nossa tela (start) quando a resolução estiver maior que 1200px.   
Então, para os breakpoints anteriores/inferiores aparecerá centralizado (center).  
  
  
Neste exemplo adicionamos o “text-md-end” que será para larguras maiores 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

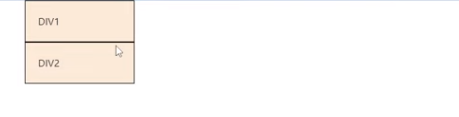
**  
**e se quisermos colocar a coluna 1 acima das colunas 2 e 3¿   
  
para isso devemos incluir uma div class=”w-100”  
  
****  
  
podemos também reordenar nossas colunas:   
  
order-last  
order-first  
  
  
  
  
Dessa forma a nossa coluna 1 ficará exposta na última posição  
  
  
podemos também 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  
****  
  
**Formatação de Colunas de Layout**