APRENDENDO FLEX BOX NA PRÁTICA

Uma forma mais intuitiva e eficiente usando teorias e exercícios para aprender FLEX BOX

ALUNOS:
José Roberto
Lucas
Vanessa
Wallysson

Passos a serem seguidos neste Aprendizado

1. Teoria:
   a. O que é o Flex Box;
   b. Para que serve e qual sua importância;
   c. Como funciona e seus principais atributos;

2. Prática:
   a. Fazer os exercícios presente no link: **https://github.com/Wallysson-utfpr/ProjFlexBox**
   b. Treinar conhecimentos adquiridos

Conceitos básicos de flexbox

- O **Flexible Box Module**, geralmente chamado de **flexbox**, foi projetado tanto como um modelo de layout unidimensional quanto como um método capaz de organizar espacialmente os elementos em uma interface.

- Quando se descreve o flexbox como sendo unidimensional, enfatiza-se o fato de que ele lida com o layout em uma dimensão de cada vez - seja uma linha ou uma coluna.

Para que serve e qual sua importância

Antes da introdução do flexbox, geralmente eram utilizadas técnicas mais complexas, como floats, position e table display para a criação de layouts. Entretanto, tais técnicas nem sempre eram fáceis de entender ou implementar, principalmente em layouts complexos.

Já com o flexbox os desenvolvedores podem **controlar a direção, o espaçamento, o tamanho e o alinhamento** dos elementos com maior facilidade e precisão.

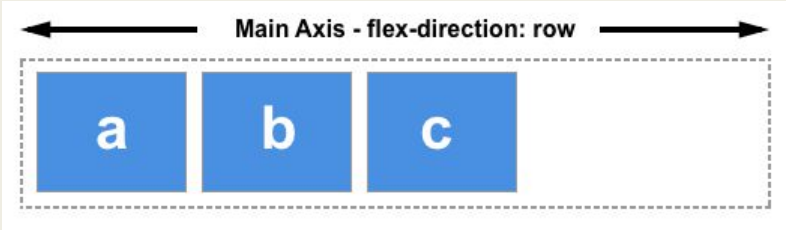
Os eixos do flexbox

- Ao se utilizar o flexbox, é preciso ter em mente que todas as operações realizadas relacionam-se a dois eixos: o **eixo principal** e o **eixo transversal**. O eixo principal é definido através da propriedade flex-direction e o eixo transversal encontra-se na direção perpendicular a ele. Como esses eixos são as engrenagens fundamentais do flexbox é necessário compreender minuciosamente o seu funcionamento.

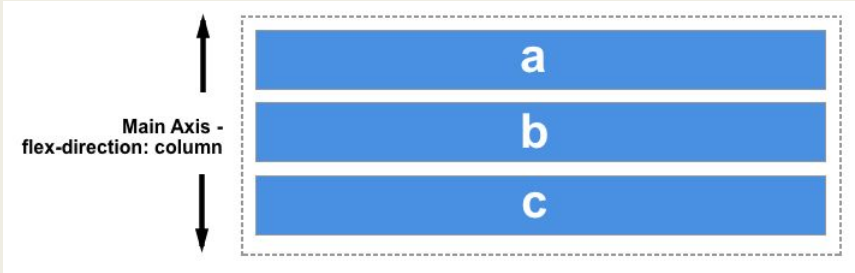
Eixo principal

- Conforme descrito acima, a propriedade **flex-direction** define a direção do eixo principal e pode ter quatro valores possíveis:
  - row
  - row-reverse
  - column
  - column-reverse

- Se o valor escolhido for row (linha) ou row-reverse (linha reversa), seu eixo principal se moverá ao longo da linha — na direção inline.



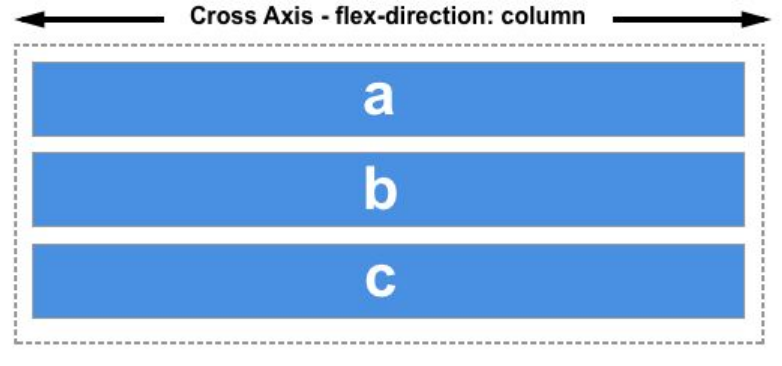
- Se o valor escolhido for column (coluna) ou column-reverse (coluna reversa) e o eixo principal se moverá do topo até o fim da página — na direção block.



Eixo transversal

- O eixo transversal é perpendicular ao eixo principal, logo, se a propriedade flex-direction estiver definida nas linhas, como row ou row-reverse, o eixo transversal estará na direção das colunas, como column ou column-reverse.

- Se o eixo principal for definido nas colunas, como column ou column-reverse, então o eixo transversal estará na direção das linhas, como row ou row-reverse.



- Compreender a diferença entre os eixos principal e perpendicular é o que importa quando começamos a observar o **alinhamento ou justificação dos itens flexíveis** (flex items); o flexbox possui propriedades que alinham e justificam o conteúdo ao longo de um eixo ou de outro.

Contêiner Flex

- A área de um documento que faz uso do flexbox é chamada de contêiner flex. Para criar essa estrutura, define-se o valor da propriedade CSS **display** do elemento representado pelo contêiner como **flex ou inline-flex**. Desse modo, os elementos-filhos deste contêiner tornam-se do tipo flex. Como todas as propriedades no CSS possuem valores padrão, ao criar um contêiner flex, os elementos nele contidos apresentarão o seguinte comportamento:
  - Exibição em linha (o padrão do **flex-direction é row**).
  - **Alinhamento na borda inicial** do eixo principal.
  - Não há expansão no eixo principal, mas pode haver contração.
  - Haverá expansão vertical para preencher a altura do eixo transversal.
  - A propriedade **flex-wrap** estará definida como **nowrap**.

Propriedade “flex-direction”

- A propriedade flex-direction permite alterar **a direção** na qual os elementos flex serão exibidos ao longo do eixo principal. Definindo a propriedade flex-direction como row-reverse (linha reversa) ainda teremos os elementos dispostos em uma linha, entretanto, a linha inicial e final serão trocadas.
- Se mudarmos a flex-direction para a column (coluna), o eixo principal exibirá os elementos em uma coluna. Por exemplo, quando o valor atribuído a essa propriedade é column-reverse (coluna reversa), a coluna inicial e final serão trocadas.

Quebra de linha com “flex-wrap”

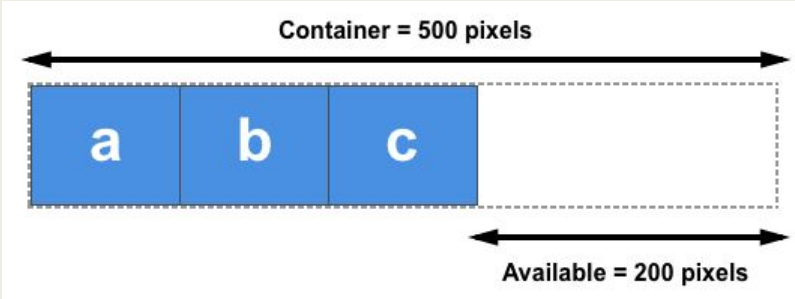
- Embora o flexbox seja um modelo unidimensional, é possível fazer com que elementos flex sejam agrupados em múltiplas linhas. Ao fazer isso, considera-se cada linha como um novo contêiner flex. Para gerar a quebra automática das linhas, basta adicionar a propriedade **flex-wrap com o valor wrap**. Assim, se elementos forem muito grandes para serem exibidos em uma única linha, eles serão agrupados em outras linhas.

Expansão, encolhimento e tamanho dos elementos flex

- Para ter mais controle sobre os elementos flex é possível alterá-los diretamente utilizando as três propriedades abaixo:
  - flex-grow
  - flex-shrink
  - flex-basis

- Antes que essas propriedades possam fazer sentido, é preciso compreender o conceito de espaço disponível. Quando se modifica o valor das propriedades acima, altera-se a forma que o espaço disponível é distribuído entre os elementos. Tal conceito de espaço disponível também é relevante quando se trata do alinhamento.

- Conforme o exemplo abaixo, se houver três elementos com 100 pixels de comprimento em um contêiner de 500 pixels, então o espaço total necessário para acomodá-los será de 300 pixels. Desse modo, sobrarão 200 pixels de espaço útil. Se os valores iniciais não forem modificados, então o flexbox posicionará esse espaço após o último item.



Se for necessário que os elementos cresçam proporcionalmente ou não e preencham o espaço disponível, deverá haver método que faça essa distribuição espacial entre eles, conforme será visto nas seções subsequentes.

Propriedade “flex-basis”

- A propriedade flex-basis define o tamanho inicial dos elementos, em unidades de pixel, antes que o espaço remanescente seja redistribuído. O valor inicial desta propriedade é alto — neste caso o navegador observa se os itens possuem o mesmo tamanho. No exemplo acima, todos os itens têm uma largura de 100 pixels, que é utilizada como padrão na propriedade flex-basis.
- Se os elementos não possuírem um tamanho padrão, então as dimensões dos seus conteúdos (imagem, texto, etc) serão passadas como parâmetro para propriedade flex-basis. É por isso que quando escreve-se display: flex no elemento-pai para criar o contêiner, todos os elementos-filhos se organizam em linha e ocupam apenas o espaço necessário para exibir seu conteúdo.

Propriedade “flex-grow”

- Com a propriedade flex-grow definida como um inteiro positivo, os elementos flex podem crescer ao longo do eixo principal, a partir do valor mínimo estabelecido no flex-basis. Isto fará com que o elemento se estique e ocupe qualquer espaço disponível nesse eixo ou uma proporção dele, caso outros elementos-irmãos também possam crescer.
- Atribuir o valor 1 à propriedade flex-grow fará com que o espaço disponível no contêiner flex seja igualmente distribuído entre todos os elementos do exemplo acima. Logo, os elementos-filhos irão se expandir para preencher o contêiner no sentido do eixo principal.
- Como visto no parágrafo anterior, a propriedade flex-grow pode ser empregada para distribuir o espaço proporcionalmente entre os elementos de um contêiner, contudo, se atribuirmos ao primeiro elemento o valor 2 para 1 aos elementos restantes, duas partes serão dadas ao primeiro elemento (100px de 200px totais) e uma parte para cada um dos outros dois elementos (50px de 200px totais).

Propriedade “flex-shrink”

- Enquanto a propriedade flex-grow permite aumentar a largura dos elementos dentro do contêiner para completar o espaço disponível no eixo principal, a propriedade flex-shrink faz o oposto, controlando a redução dos mesmos. Caso não haja espaço suficiente para acomodar todos os elementos e o valor da propriedade flex-shrink seja um inteiro positivo, a largura pode ser reduzida a um valor menor do que a definida na propriedade flex-basis. Assim como na propriedade flex-grow, diferentes valores podem ser atribuídos a um elemento de modo que ele encolha mais do que os outros - um elemento cuja propriedade flex-shrink receba um valor inteiro maior irá diminuir mais do que os seus irmãos que tenham valores menores.

Alinhamento, justificação e distribuição de espaço livre entre os elementos

- Um atributo chave do flexbox é a capacidade de alinhar e justificar os elementos flex nos eixos principal e transversal e distribuir o espaço entre eles.

Propriedade align-items

- A propriedade align-items irá alinhar os elementos no eixo transversal.
- O valor inicial desta propriedade é stretch e é por essa razão que, por padrão, os elementos flex se estendem até a maior altura. De fato, eles se esticam para preencher o contêiner flex - o item mais alto define a altura deste.
- Pode-se definir a propriedade align-items como flex-start, de modo que os elementos fiquem alinhados com o topo do contêiner, flex-end para alinhá-los a partir da base ou center, para que o alinhamento seja centralizado.
- Teste essa propriedade e seus possíveis valores no exemplo prático abaixo — colocou-se uma determinada altura no contêiner flex, de modo que se perceba como os elementos podem ser movidos no interior do mesmo. Veja o que acontece ao definir cada um dos possíveis valores da propriedade align-items:
  - stretch
  - flex-start
  - flex-end
  - center

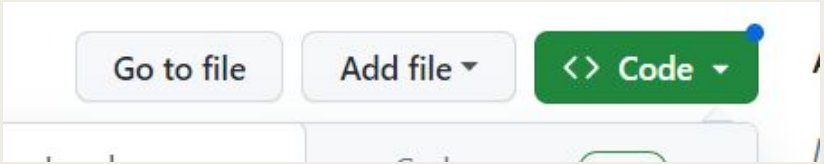
Propriedade justify-content

- A propriedade justify-content é empregada para alinhar os elementos ao longo do eixo principal, cuja direção (row ou column) é definida a partir da propriedade flex-direction. O valor inicial é flex-start, que alinha os elementos rente à borda esquerda do contêiner, mas também pode ser definido como flex-end, que resulta em um alinhamento oposto, rente à borda direita do contêiner, ou center, para alinhá-los ao centro.
- O valor space-between pode ser usado pode ser usado para ocupar todo o espaço livre após a disposição dos itens e dividi-lo igualmente entre os itens, para que haja a mesma quantidade de espaço entre cada elemento. Para gerar uma quantidade igual de espaço à direita e à esquerda, usa-se o valor space-around.
- Tente os seguintes valores da propriedade justify-content quando for treinar com os exemplos práticos:
  - stretch
  - flex-start
  - flex-end
  - center
  - space-around
  - space-between

Exercícios

https://github.com/Wallysson-utfpr/ProjFlexBox

1º Acesse o repositório e clona o projeto clicando na opção em verde e escolhendo o modo para clonar



2º Abra o projeto e encontre o arquivo “index.html”

3º Clique para abrir em algum navegador e faça os exercícios

Treinamento

- AGORA É A SUA VEZ.

- Crie um repositório novo e reproduza a tela principal do Discord usando FlexBox na área de navegação, conteúdo textual e botões.