FlexBox – Alura

1. **Aula 1 – Introdução ao FlexBox e fazendo o cabeçalho:**
   1. Flexbox trata principalmente sobre realocação e posicionamento de elementos do nosso site por CSS, deixando tudo mais organizado e melhor construído.
   2. O ideal, para que não fique muita coisa no css de estilização é criar um novo arquivo chamado flexbox onde você irá fazer todas as modificações de posicionamento.
   3. Podemos começar colocando o cabeçalho e navegação em display: inline-block;
   4. Dando um *vertical-align: Middle;* os nossos elementos se alinham no centro verticalmente.
   5. Porém, tudo isso que foi visto, não é uma boa prática, pois precisamos utilizar “números mágicos” para alinhar da maneira que queremos, sendo eles números que não tem embasamento, colocamos pq achamos que ficou bom.
   6. A primeira coisa a se fazer quando vamos usar flexbox, é pensar quais são os elementos que queremos alinhar e posicionar.
   7. O melhor a se fazer é colocar o *display: flex;* no elemento mãe de tudo o que você quer fazer o posicionamento.
   8. Setando isso, podemos usar o *align-itens: center;* para alinhar todos os itens desse elemento mãe verticalmente.
   9. Podemos também colocar o espaço de todo o contêiner mãe entre os dois elementos filhos dela usando o comando *justify-content: space-between;* dessa forma, jogando os dois elementos de um cabeçalho para as laterais, por exemplo.
   10. Para alinhar itens de uma barra de navegação verticalmente, basta colocar como *display: flex;* e ele irá alinhar automaticamente.
   11. Nem todos os navegadores podem aceitar muito bem o flexbox. O site: <https://caniuse.com> diz quais são as ferramentas que você pode utilizar em quais navegadores.
2. **Aula 2 – Fazendo o footer e controlando melhor os elementos:**
   1. Podemos colocar espaçamento em volta dos elementos com *justify-content: space-around;* para deixar os elementos mais disbribuidos.
   2. O *display: flex;* também faz com que os filhos de um elemento fiquem todos com a mesma altura, facilitando na hora da estilização.
   3. Podemos selecionar a direção que queremos que o flex seja colocado. Ele por padrão vem na horizontal (*flex-direction: row;*), mas se utilizarmos o *flex-direction: column;* todos os elementos ficarão um abaixo do outro.
   4. Porém, o problema de colocar em colunas é que mesmo que você defina o tamanho do elemento pai, ele vai seguir para baixo indefinidamente, contudo, podemos utilizar o *flex-wrap: wrap;*, que basicamente diz que quando chegar no tamanho do elemento pai ele irá quebrar a linha e continuar na próxima.
   5. Além disso, podemos utilizar o *flex-flow: colunm wrap;* e, desse modo, substitui os dois comandos acima, pois ele faz as duas coisas.
3. **Aula 3 - Grid principal e limitações do flexbox:**
   1. Podemos fazer um grid utilizando os conhecimentos que já temos de flex. Basta colocar todos os itens de uma lista como display: flex;, definir um tamanho para os elementos da lista, caso estejam muito aglomerados e, por fim, colocar o espaçamento entre os itens, dando uma margem para eles.
   2. O problema é que quando não se tem todos os elementos necessários para deixar o grid 100% completo, fica um vão entre os itens da grid, onde os 2 itens são jogados para os cantos, fugindo da ordem de ser um do lado do outro no caso onde tem 4 elementos na linha do grid.
   3. Para concertar isso, nós infelizmente precisaremos utilizar contas e retirar o espaçamento between do justify-content.
   4. Para selecionar apenas itens múltiplos de algum número e fazer uma estilização somente nesses elementos de uma lista no grid, por exemplo, utilizamos o .*nomeDaClasse:nth-child(4n){config}*, nesse caso, a cada 4 elementos essa configuração será adicionada.