## Flexbox: Posicione elementos na tela

- Esse curso capacita a:
    - Aprender a famosa especificação flexible box para posicionar elementos na página;
    - Entender as diversas propriedades do flexbox e como usá-las;
    - Entender como as propriedades do flexbox substituem float, inline e inline-block;
    - Elaborar um site responsivo com flexbox;

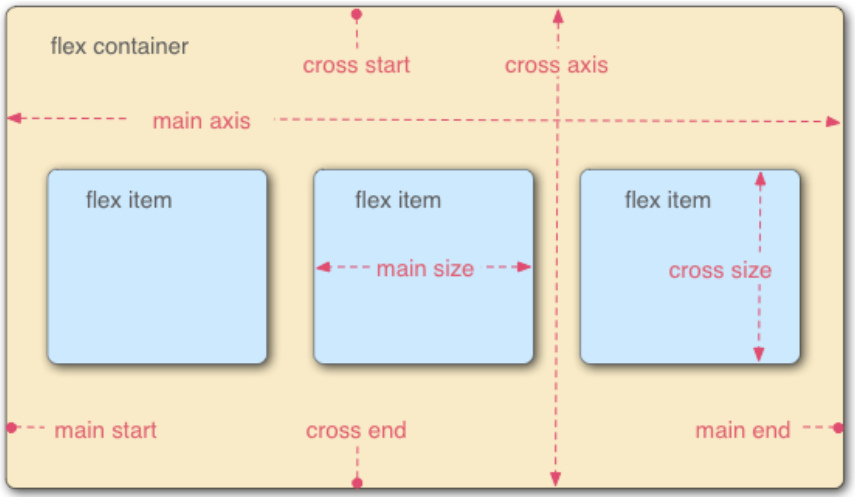
Aulas:

1. **Introdução ao flexbox e fazendo o cabeçalho:**
    - **display: inline** → Colocando nos elementos permite eles se posicionarem um do lado do outro, o problema é que os elementos não aceitam mais que seja modificada tanto a **width** quanto a **height**;
    - **display: inline-block** → Permite os elementos se posicionarem um do lado do outro porém, diferentemente do **inline** ele permite que os elementos tenha sua **width** e **height** modificadas. O problema de usar ele é espaçar os elementos entre si. Para fazer isso teríamos que colocar **margins** e fazer contas;
    - **float: left | right** → Empurra o elemento para um dos lados (left / right) e os elementos que estão embaixo sobem. Não permite o uso da propriedade **vertical-align**: **middle** para alinhar os elementos verticalmente. Ou seja, para contornar isso uma possibilidade seria ter que colocar **margin-top** ou **bottom** nos elementos e usar números mágicos;
    - **display: flex** → Permite os elementos ficarem um do lado do outro, permite espaçar os elementos de forma mais intuitiva e sem ter que fazer cálculos. Além de permitir alinhar os elementos verticalmente de forma fácil;

2. **Fazendo o footer e controlando melhor os elementos:**
    - **justify-content: flex-end** → Coloca todo espaço à esquerda, jogando o conteúdo para direita;
    - **justify-content: flex-start (default)** → Coloca todo espaço à direita, jogando o conteúdo para esquerda;
    - **justify-content: center** → Coloca todo espaço à esquerda e à direita, jogando o conteúdo para o meio;
    - **justify-content: space-between** → Coloca todo espaço entre os elementos;
    - **justify-content: space-around** → Coloca o espaço em volta dos elementos;
    - **flew-direction: column // flex-wrap: wrap;**

3. **Grid principal e limitações do flexbox**

4. **Arrumando os elementos com flex para mobile:**
    - Para mobile devemos colocar um elemento em baixo do outro, é a melhor forma de ocupar todo o espaço para melhorar a usabilidade no celular. Para isso pode-se colocar a propriedade **flex-direction: column**, que faz com que os elementos fiquem um em baixo do outro;




    - **flex container**:
        - display: flex;
        - flex-direction;
        - justify-content;
        - flex-wrap;
        - flex-flow;
        - align-items;
        - align-content;
    - **flex item:**
        - order;
        - flex-grow;
        - flex-shrink;
        - flex-basis;
        - flex;
        - align-self;

5. **Mais sobre flexbox & desafios:**
    - **flex-grow** → Um elemento (flex item) cresce e ocupa todo o espaço que está sobrando dentro do flex container, para isso, o valor 1 é suficiente;
    - **flex-shrink** → Define que um elemento não diminua;
    - A propriedade **flex-basis** serve para definir uma largura ou altura para o flex item. Se o flex container tiver com flex-direction: column, o **flex-basis** no flex item servirá para definir uma height. Caso o flex-direction: row, ele funciona como um width;