CURSO DESENVOLVIMENTO WEB / CSS

Guia do Flebox

CURSO DESENVOLVIMENTO WEB
André Fontenelle

https://www.udemy.com/course/curso-desenvolvimento-website-responsivo-completo/

O que vamos aprender

- Definição de FlexBox.
- Compreensão das propriedades.
- Propriedades para o elemento pai.
    - Display.
    - Flex Direction.
    - Flex Wrap.
    - Flex Flow.
    - Justify-content.
    - Align-items.

- Propriedades para o elemento filho.
    - Order.
    - Flex-Grow.
    - Flex-Shrink.
    - Flex-Basis.
    - Align-self

- Bibliografia.
- Tutoriais no YouTube.

FlexBox

Definição

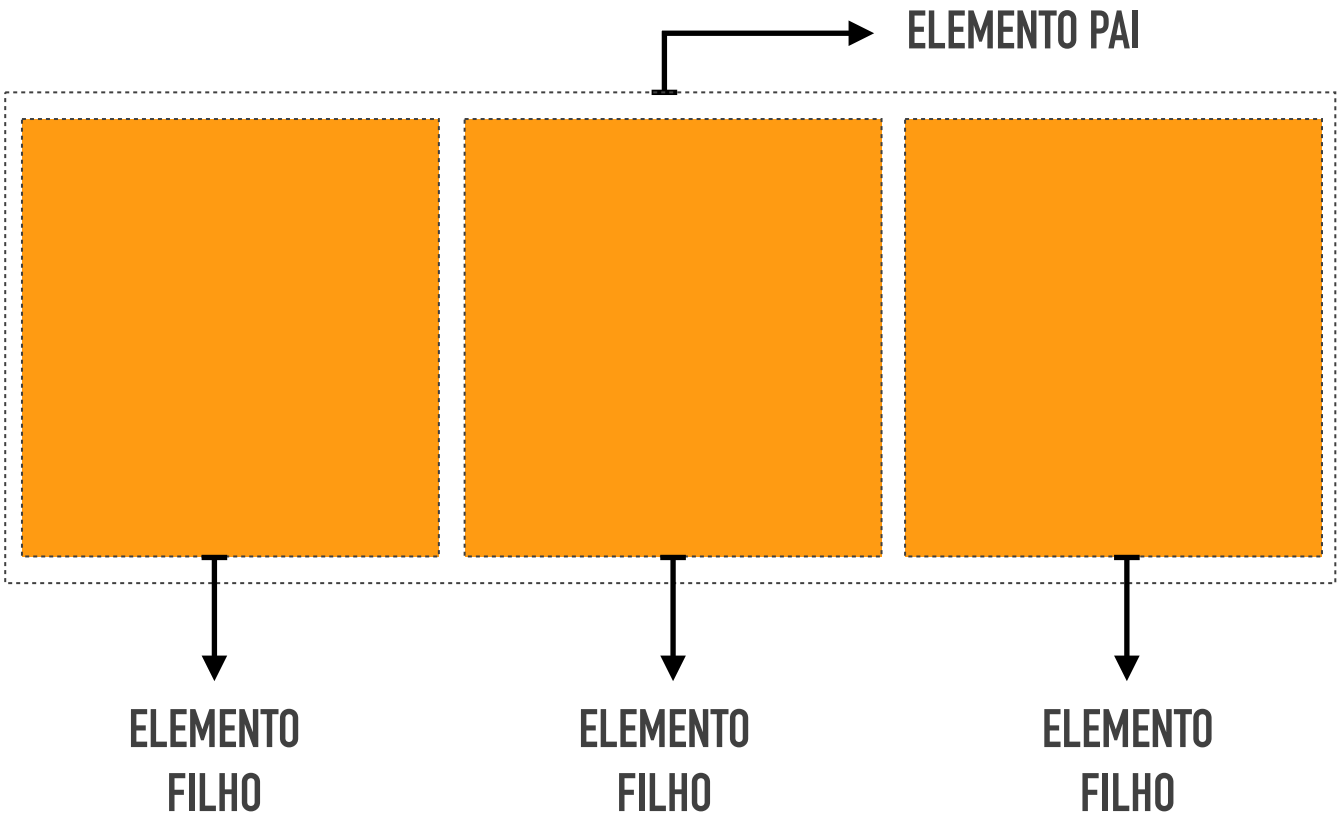
O **Flexible Box Module**, geralmente chamado de **FlexBox**, foi projetado tanto como um modelo de layout unidimensional quanto como um método capaz de organizar espacialmente os elementos em uma interface, além de possuir capacidades avançadas de alinhamento. Este artigo fornece um resumo das principais funcionalidades do **FlexBox**, as quais exploraremos com mais detalhes no restante deste guia.

O **FlexBox** é o fim da dor de cabeça para organizar os elementos de layout. Era comum o webdesigner perder horas corrigindo o posicionamento entre os elementos, seus espaços e alinhamento. O FlexBox torna tudo isso muito simples e fácil.

FlexBox

Para fácil compreensão do uso do FlexBox, o webdesigner precisa ter claro em sua mente, as propriedades aplicadas no elemento pai e nos elementos filhos.

As propriedades no elemento pai, são os mais importantes e comuns de serem aplicados. Muitas vezes, sequer precisa-se aplicar propriedades de FlexBox nos elementos filhos.



Propriedades para o elemento pai.

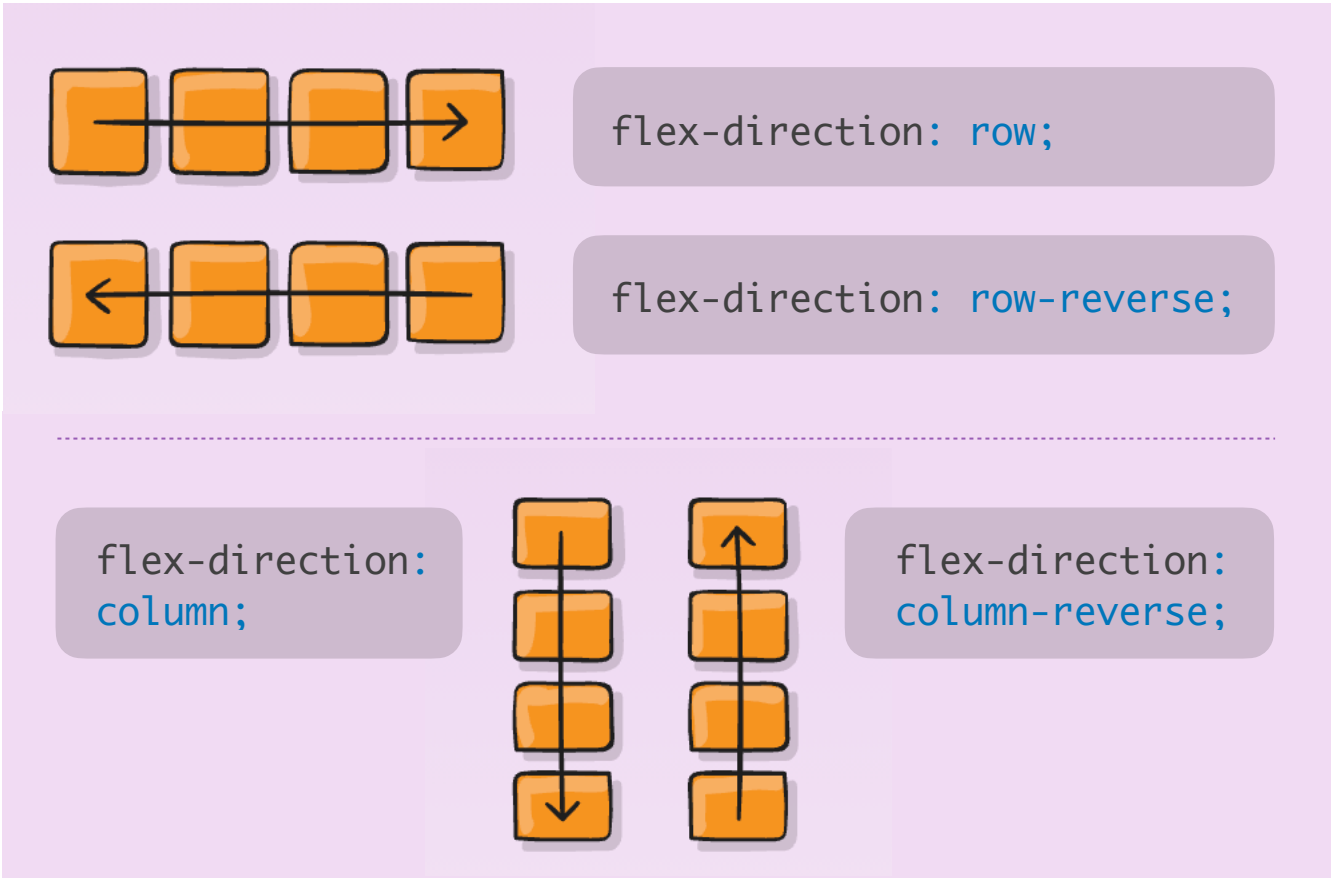
Display

Isso define um flex container. Ele permite um contexto flexível para todos os elementos filhos diretos.

```
display: flex;
```

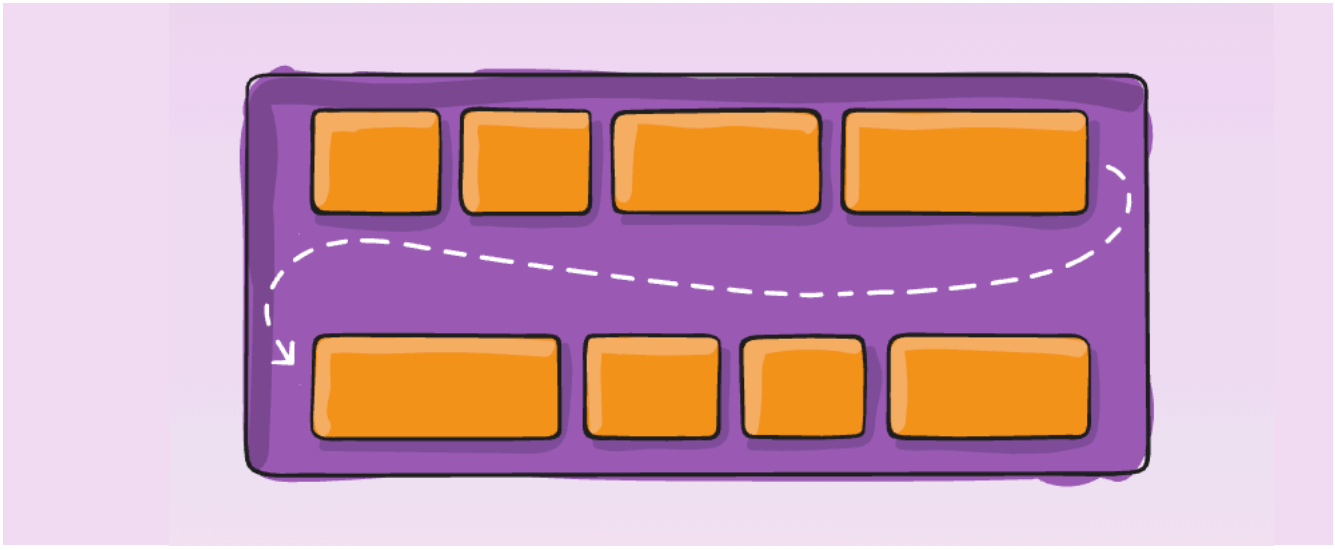
Flex Direction

Esta propriedade define a principal forma de distribuição dos elementos em coluna ou linha.



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. Qualquer distribuição espacial ocorrerá ao longo essa linha, sem referência às linhas de ambos os lados. Para gerar a quebra automática das linhas adicione 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.



Flex-Flow

A propriedade Flex-Flow é a união das duas propriedades mencionadas anteriormente. Um atalho para usar mais rápido as propriedades **flex-direction** e o **flex-wrap**.

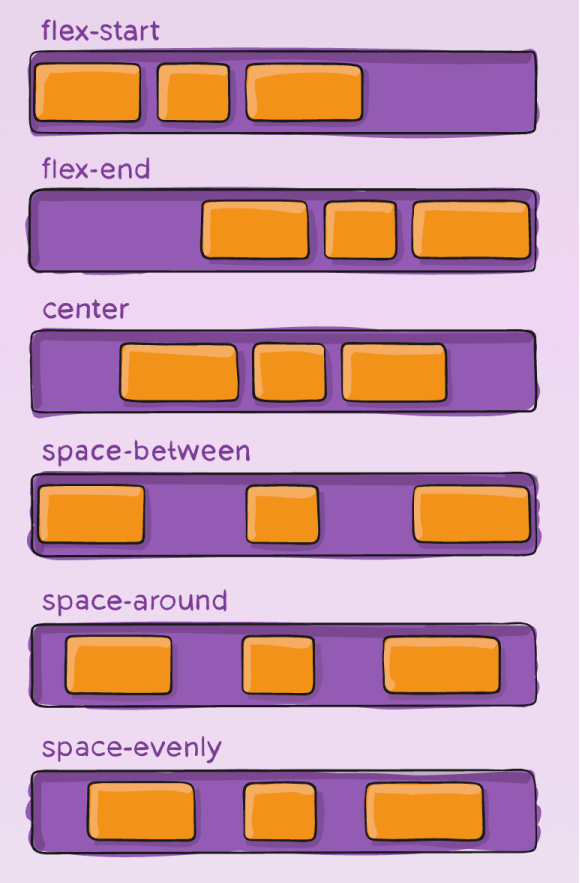
```
flex-flow: row wrap;
```

```
flex-flow: column wrap;
```

Justify-content

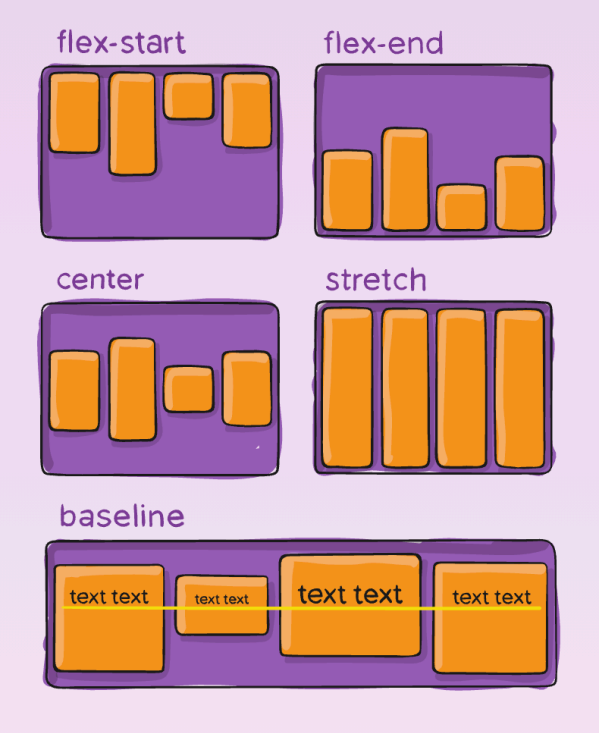
A propriedade justify-content é empregada para alinhar os elementos ao longo do eixo principal. O valor inicial é flex-start, que alinha os elementos rente à borda esquerda do contêiner, mas também pode ser definido como flex-end, que resulta em um alinhamento oposto, rente à borda direita do contêiner, 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.



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 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 (acima) o que acontece ao definir cada um dos possíveis valores da propriedade align-items.

Propriedades para o elemento filho.

Order

Por padrão, os itens são organizados pela ordem que estão no código. Entretanto, a ordem dos elementos podem ser alterados utilizando a propriedade order.

```
section.amarelo {
     order: 3;
     background-color: yellow;
}

section.verde {
     order:1;
     background-color: green;
}

section.azul {
     order:2;
     background-color: blue;
}
```

```
<div>
    <section class="amarelo">1</section>
    <section class="verde">2</section>
    <section class="azul">3</section>
</div>
```

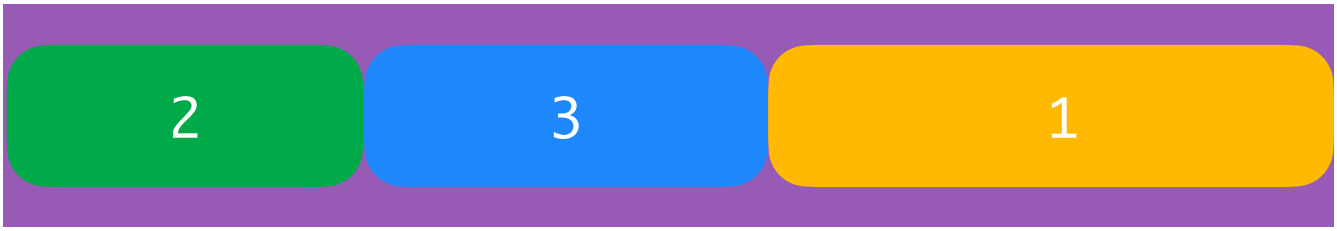
Flex-grow

Estabelece qual elemento filho terá um crescimento diferente dos demais. O valor padrão é 0 (zero). Ao determinar um dos elementos filhos com valor 1, este elemento assumirá um tamanho maior. Se um terceiro elemento, assume valor 2 para a propriedade **flex-grow**, este elemento terá um tamanho maior ainda.

```
section.amarelo {
     order: 3;
     background-color: yellow;
     flex-grow:2;
}

section.verde {
     order:1;
     background-color: green;
     flex-grow:0;
}

section.azul {
     order:2;
     background-color: blue;
     flex-grow:1;
}
```



Flex-shrink

Estabelece qual elemento filho terá um redução diferente dos demais. É a propriedade inversa ao flex-grow. Quanto maior for o valor atribuído, maior será a redução. Importante, há uma condição do flex-shrink funcionar, o flex-wrap não pode estar como valor wrap. Precisa estar como **nowrap** ou não ser definido.

Dica para memorizar:
Quanto menor o valor atribuído, menos vai reduzir seu tamanho.

Flex-basis

É útil para estabelecer elementos elástico. Usando a propriedade flex-basis, não é necessário determinar a largura dos elementos.

É necessário usar o flex-grow para usar o flex-basis.

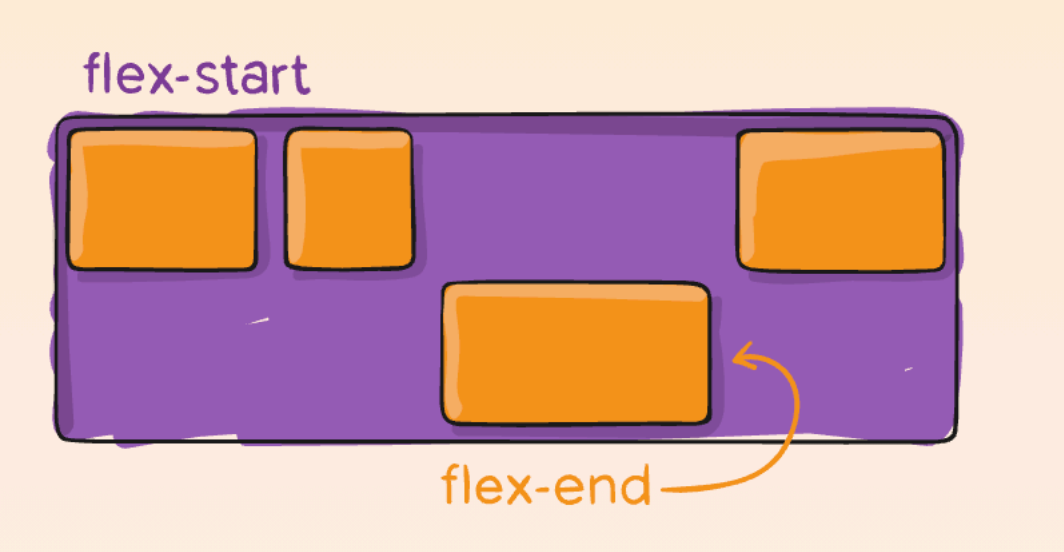
Atalho

Pode ser substituído por:

flex: 1 auto;

Align-self

Permite o alinhamento dos elementos (ou de algum elemento específico). Empurrando o elemento para o fim do elemento pai.



Bibliografia

A Complete Guide to FlexBox

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Conceitos básicos de FlexBox

https://developer.mozilla.org/pt-BR/docs/Web/CSS/CSS_Flexible_Box_Layout/Conceitos_Basicos_do_Flexbox

Tutoriais no YouTube

Tutorial - Fundamento e Prática de FlexBox

https://youtu.be/VuivAZVLIlY