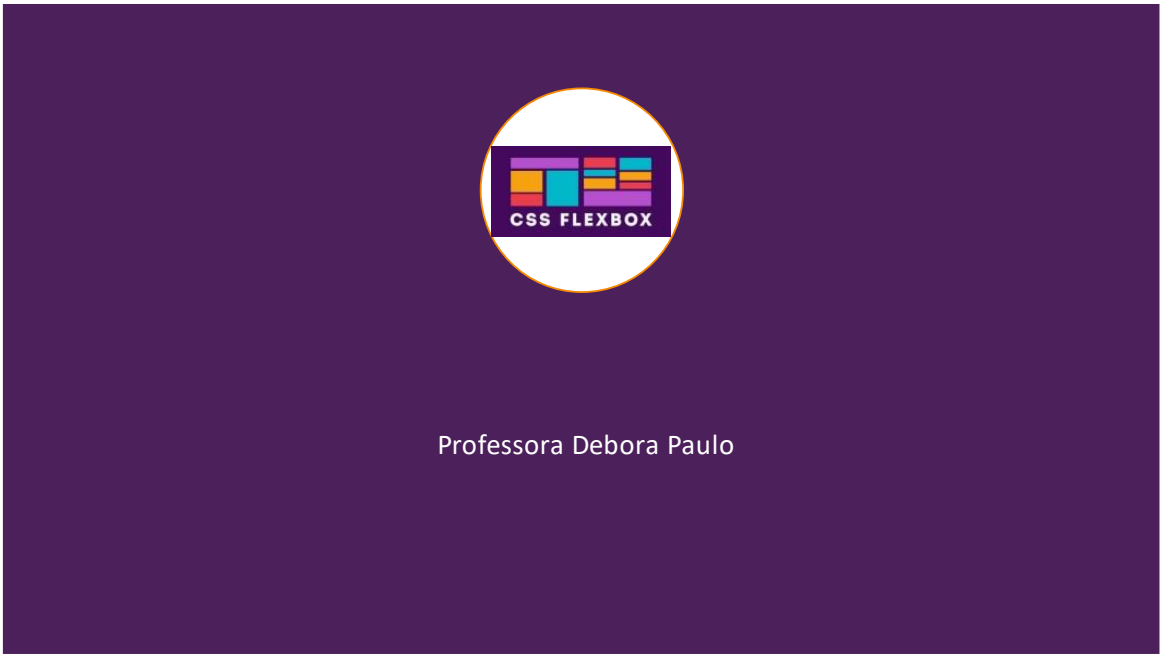
Etec Antônio Furlan Barueri

Centro Paula Souza

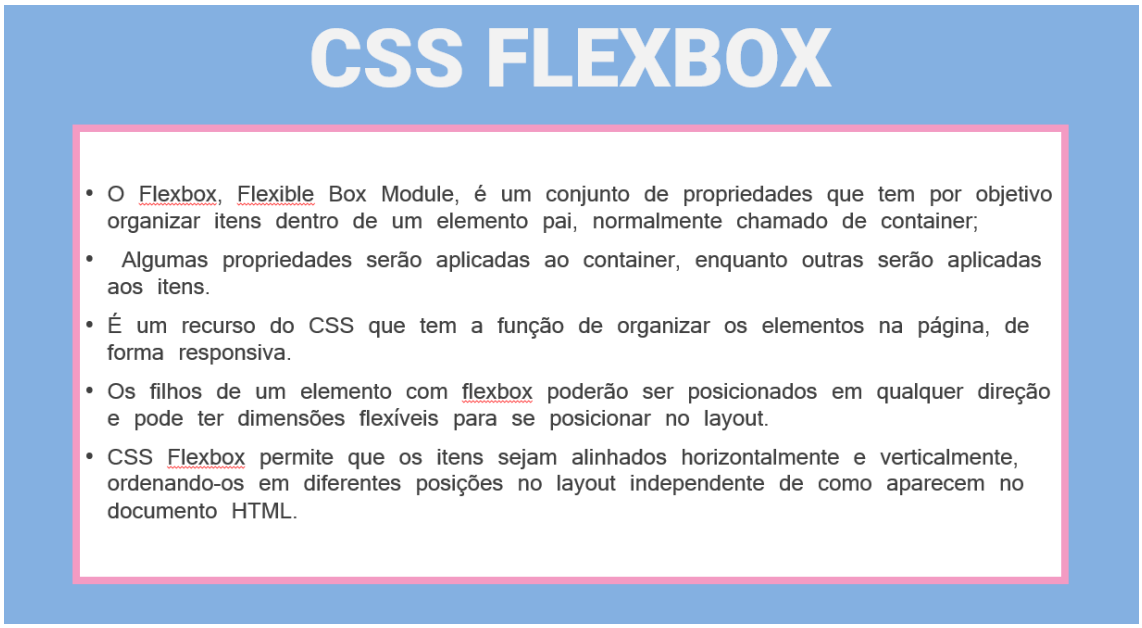
Governo do Estado São Paulo

Professora Debora Batista Paulo



Conceitos Iniciais:

Durante um longo período, as únicas formas para criar leiautes em CSS e posicionar elementos que fosse compatível entre browsers eram float e position. Essas ferramentas disponíveis possuíam limitações especialmente no que tange ao requisito de responsividade. Sendo assim, o Flexbox (Flexible Box) foi criado para tornar essas tarefas mais simples e funcionais: os filhos de um elemento com Flexbox podem se posicionar em qualquer direção e pode ter dimensões flexíveis para se adaptar.



DESIGN RESPONSIVO COM FLEXBOX

É um recurso do CSS que tem a função de organizar os elementos na página de forma responsiva, flexível.
Os filhos de um elemento com flexbox vão poder se posicionar em qualquer direção e pode ter dimensões flexiveis para se adaptar.

Para iniciarmos, vamos criar um layout simples:

```
<body>
    <section class="container">
        <article class=" item article-01">
            Artigo 01
        </article>

        <article class=" item article-02">
            Artigo 02
        </article>

        <article class=" item article-03">
            Artigo 03
        </article>
    </section>

</body>
```

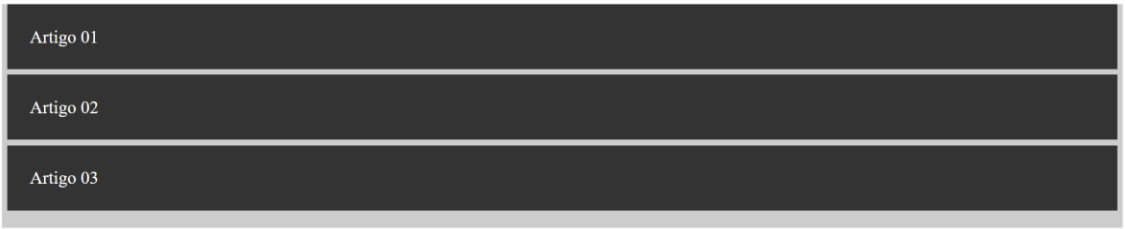
CSS

```
.container{
background-color: #ccc;
height: 200px;
```

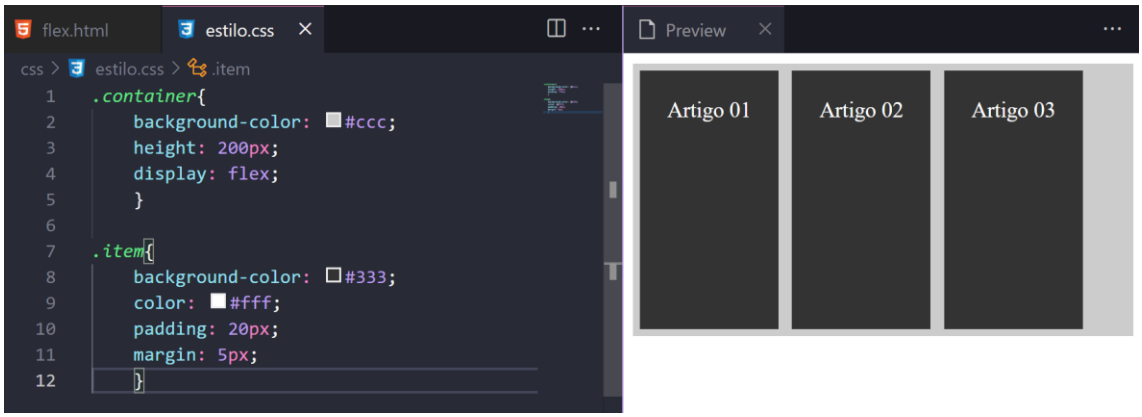
```
}

.item{
background-color: #333;
color: #fff;
padding: 20px;
margin: 5px;
}
```

RESULTADO:



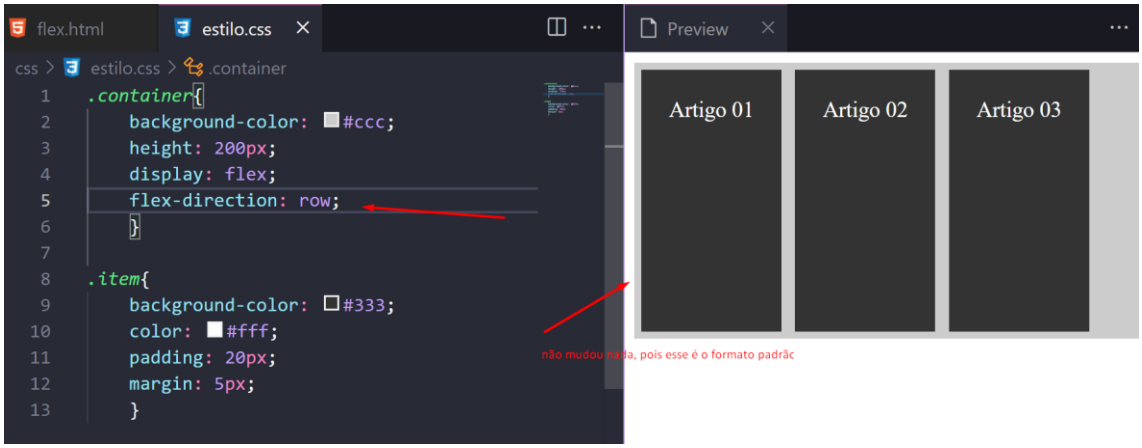
Agora vamos adicionar ao elemento container, a propriedade display: flex;



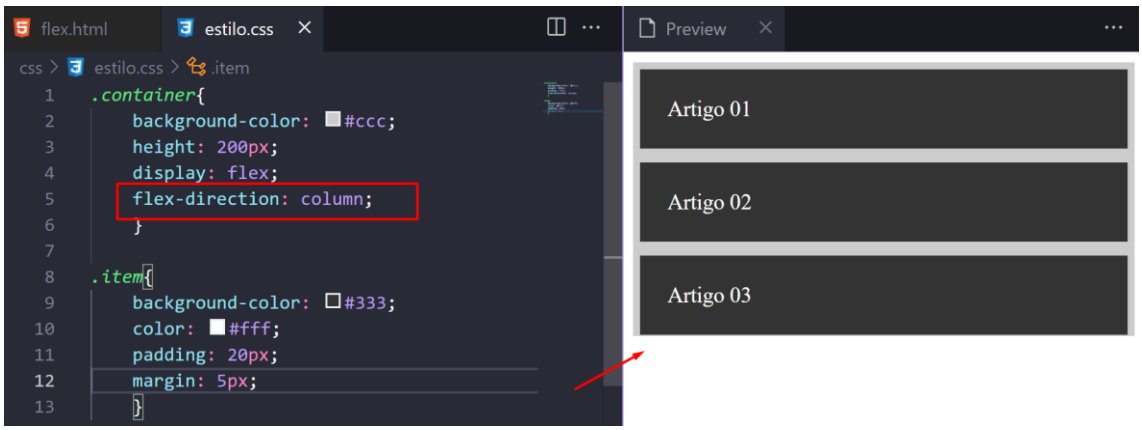
Ao adicionarmos a propriedade flex ao container, elemento pai, outras propriedades automaticamente são adicionadas a ele.

FLEX DIRECTION:

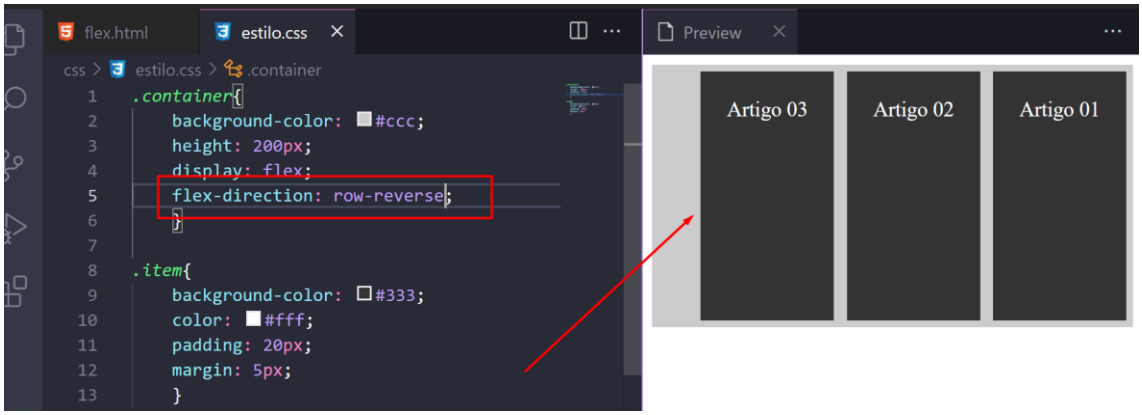
Essa propriedade define a direção, que por padrão assume o valor de row, fazendo com que os itens fiquem em formato de linha.



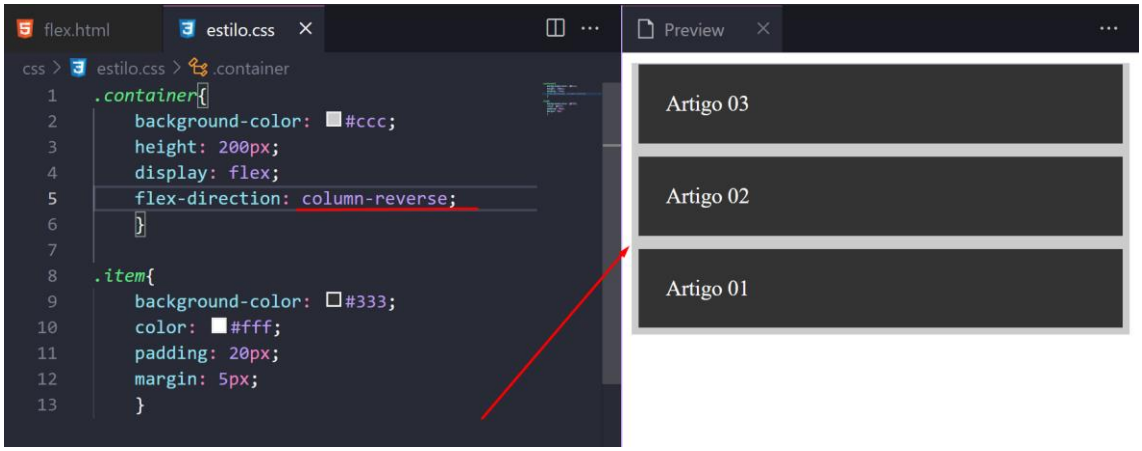
Além do **row**, temos o **column** que define os itens em um formato de coluna.



Temos também o **row-reverse**, que muda a direção dos itens da direita para a esquerda.

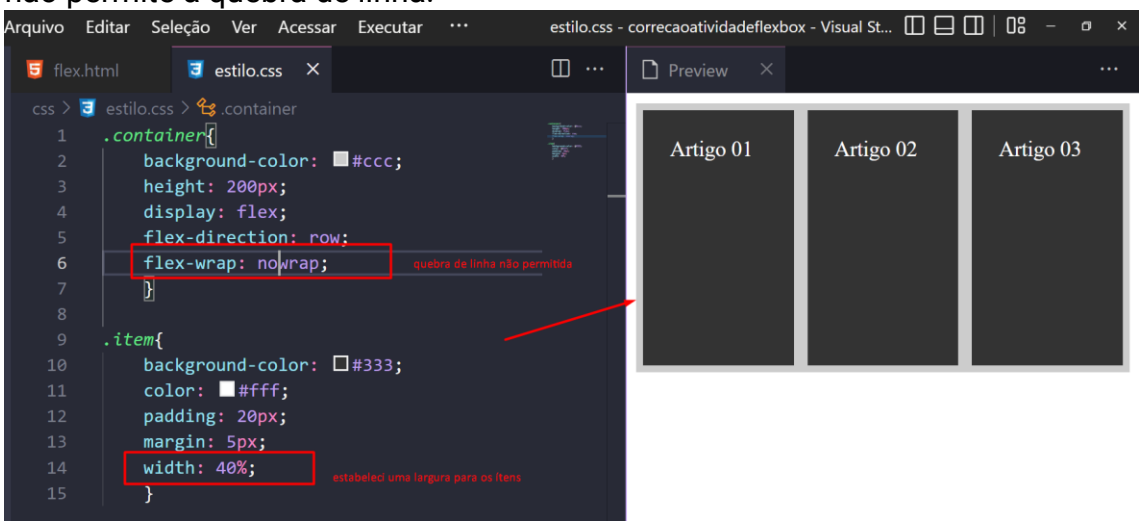


O **column-reverse**, onde os itens são renderizados para o formato coluna direção reversa.

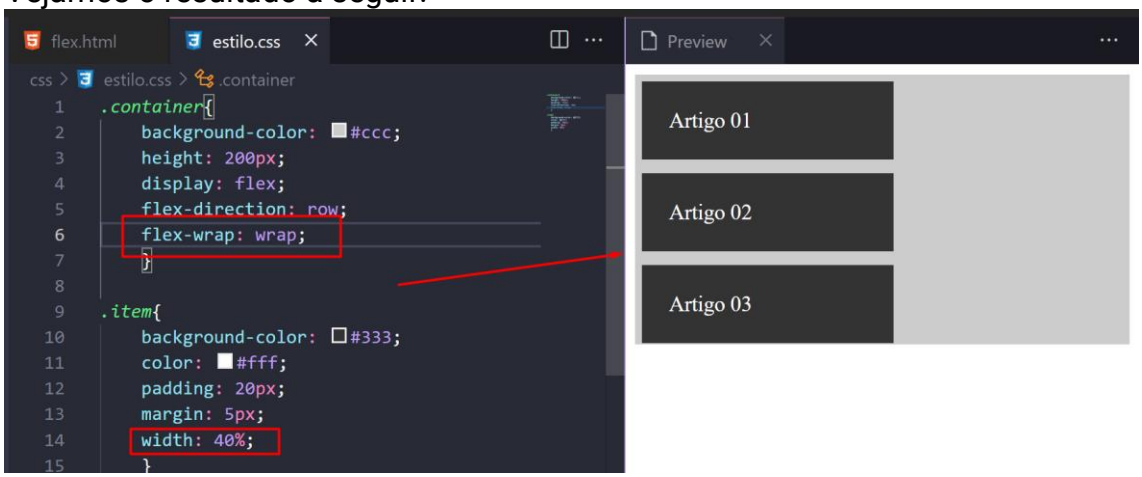


FLEX-WRAP

Além do flex Direction, temos também o flex-wrap que irá permitir se irá ou não, ter quebra de linha.
No exemplo abaixo, estamos definindo um flex-wrap: nowrap, que por padrão não permite a quebra de linha.

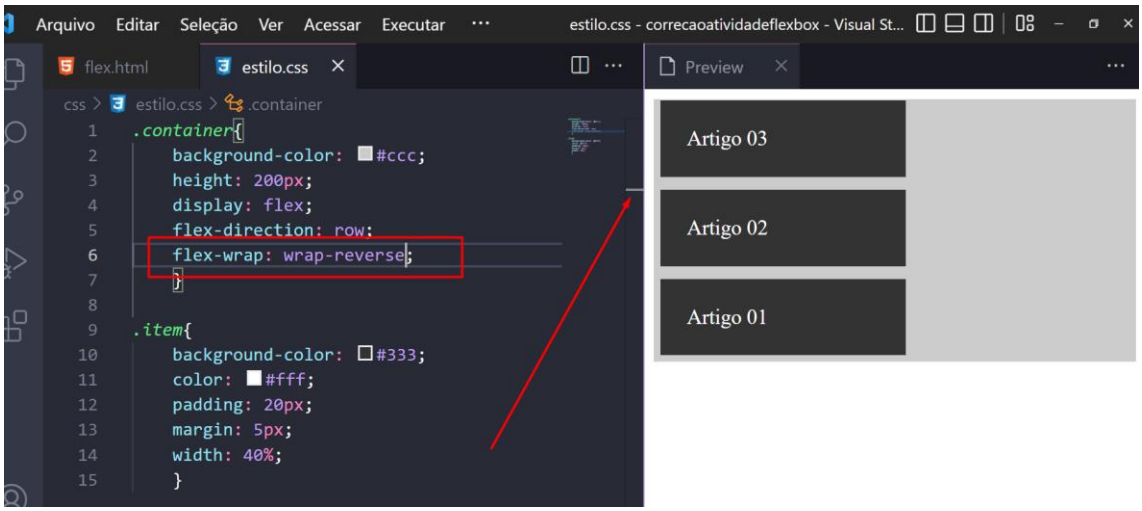


Já no formato de flex-wrap: wrap, estamos permitindo a quebra de linha. Vejamos o resultado a seguir:



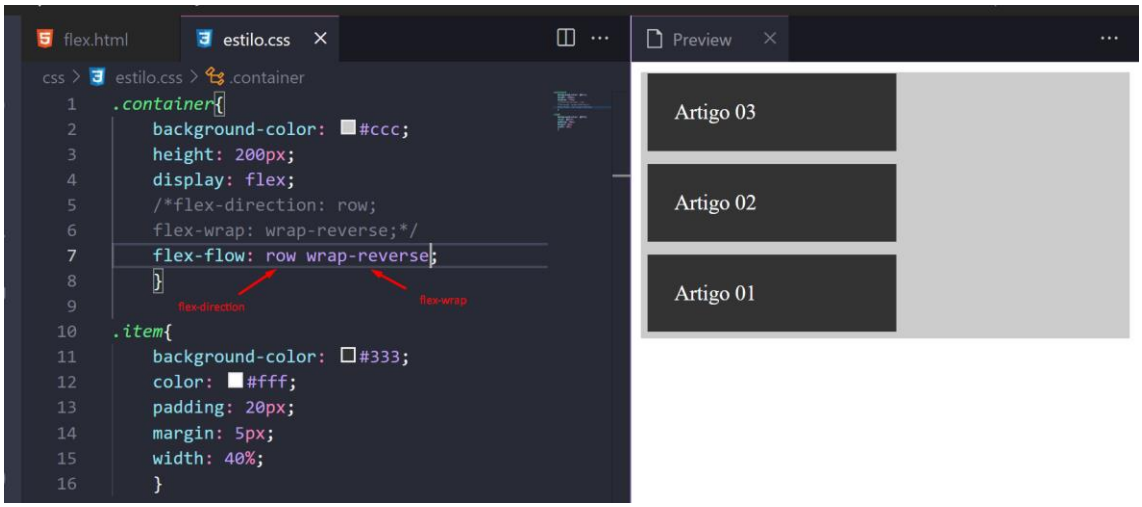
Visualização no browser:

Tem ainda o **wrap-reverse** que renderiza de forma reversa.



FLEX-FLOW

O flex-flow é a forma resumida para atribuir as propriedades flex-direction e flex-wrap. Escrevemos em uma única linha.
Por exemplo:
flex-flow: row wrap-reverse

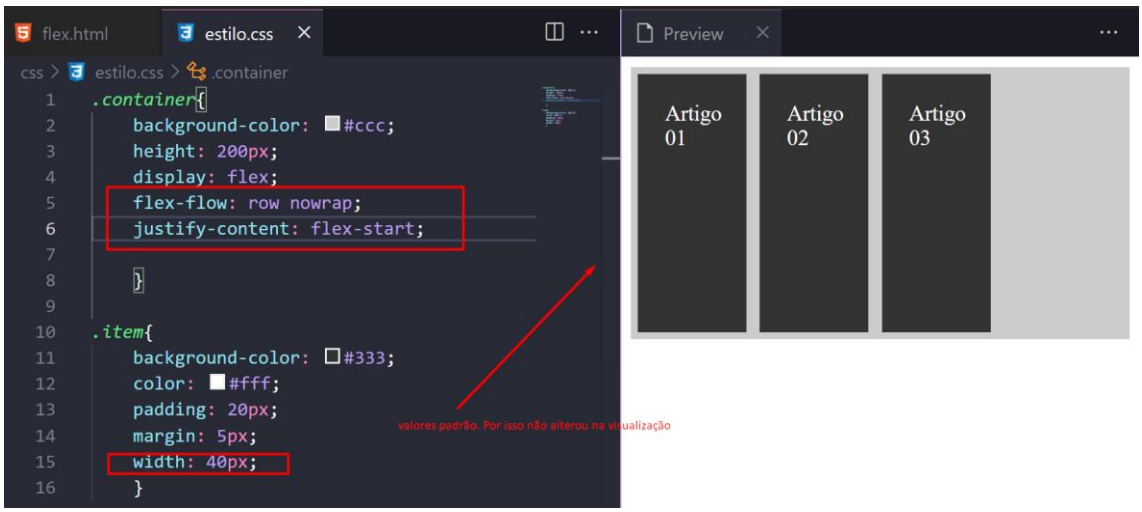


JUSTIFY-CONTENT

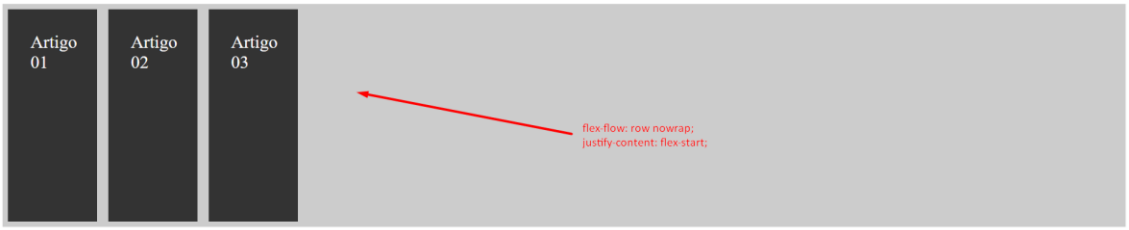
Faz o alinhamento horizontal. Seu valor padrão é o flex-start.

FLEX-START

Alinha os itens da esquerda para direita;

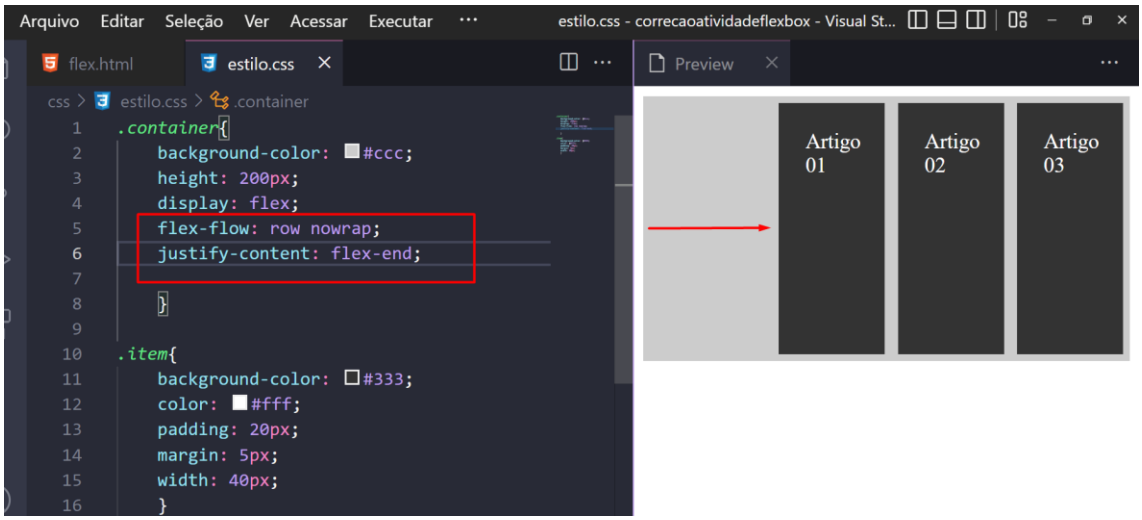


Visualização no browser:



FLEX-END:

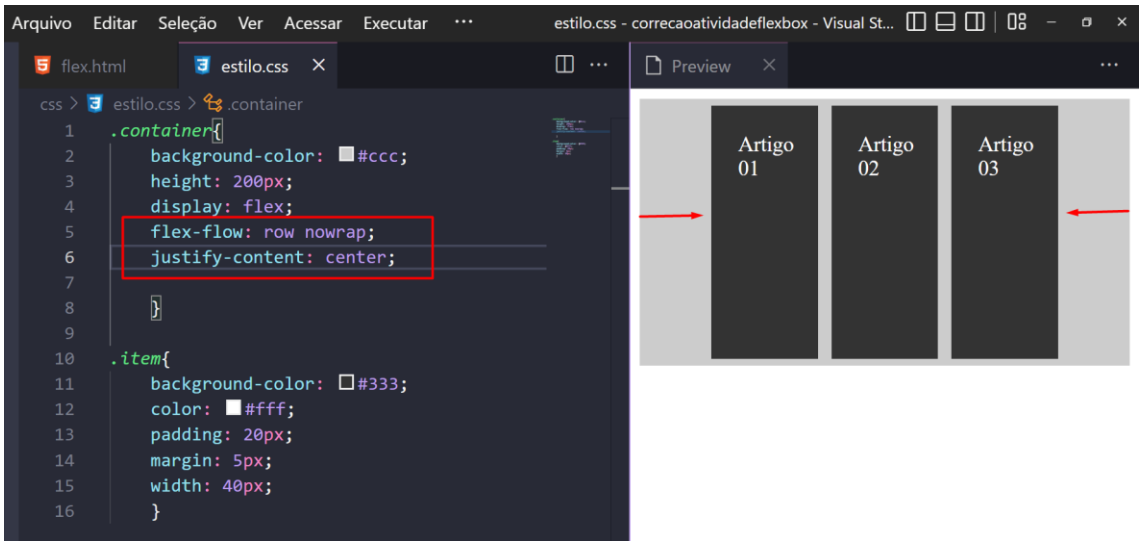
Alinha os itens da direita para esquerda não alterando a ordem dos elementos;



Visualização no browser:

CENTER:

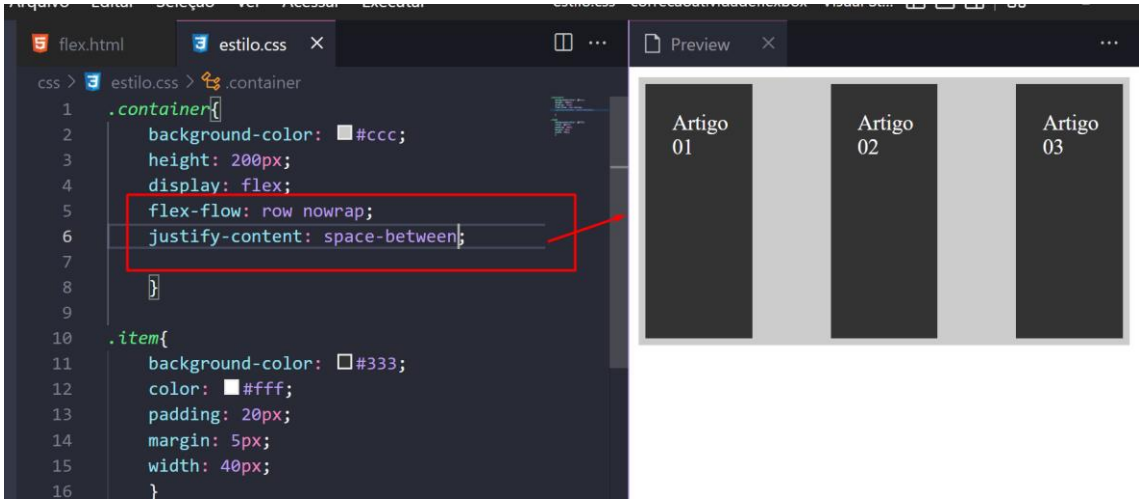
Faz o alinhamento de forma centralizada.



Visualização no browser:

SPACE-BETWEEN:

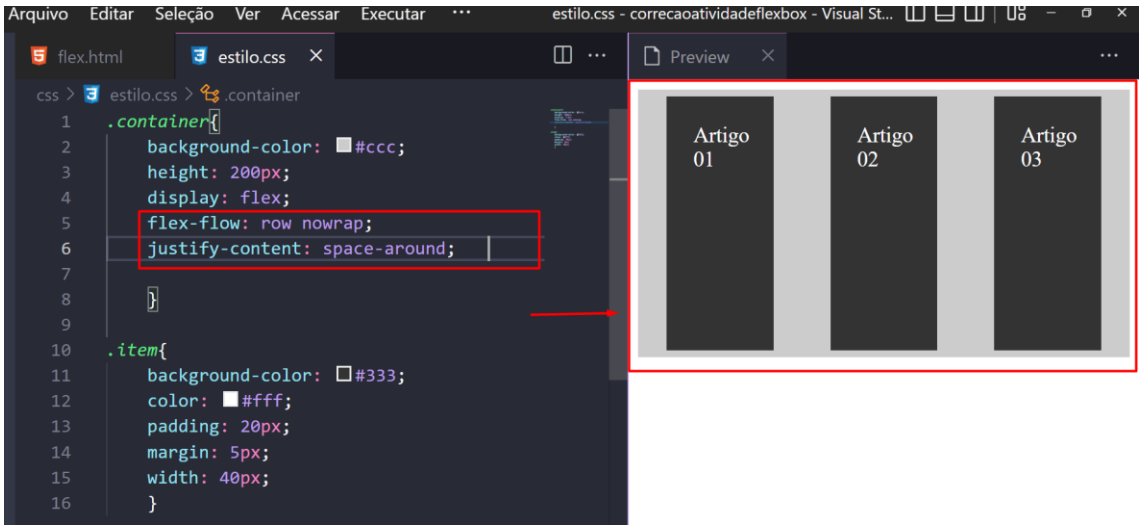
Esta propriedade cria um espaço entre os itens de forma responsiva, todos alinhados.



Visualização no browser:

SPACE-AROUND:

Esta propriedade também cria um espaço entre os itens de forma responsiva, inclusive ao redor mantendo o alinhamento na página.

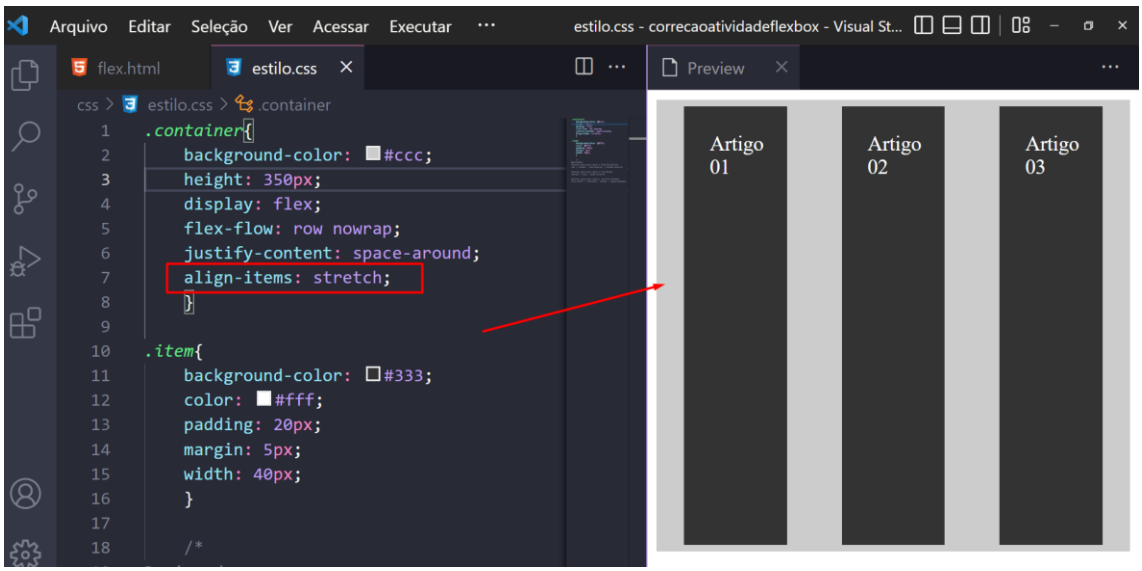


Visualização no browser:

ALIGN-ITENS

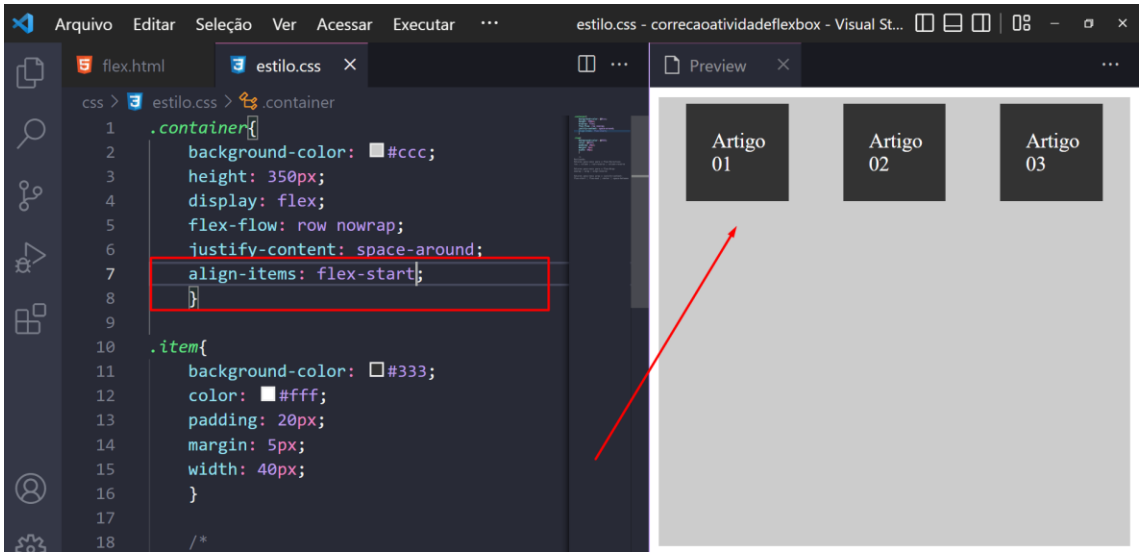
É o responsável por fazer o alinhamento vertical. O valor padrão é o stretch que alinha de forma igual na vertical.

STRETCH



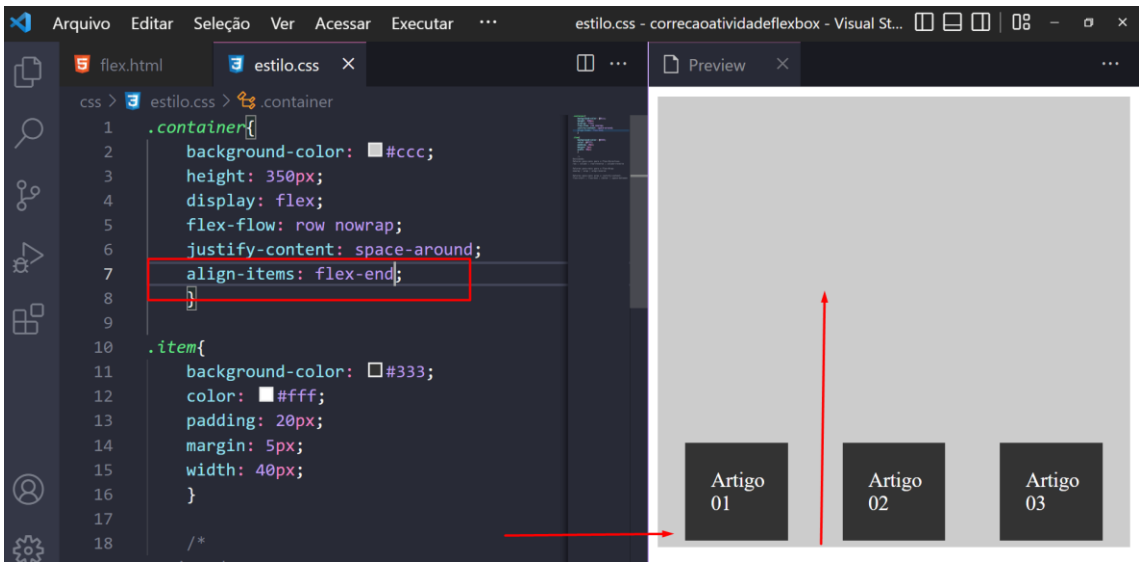
FLEX-START:

Faz o alinhamento dos itens no topo à vertical.



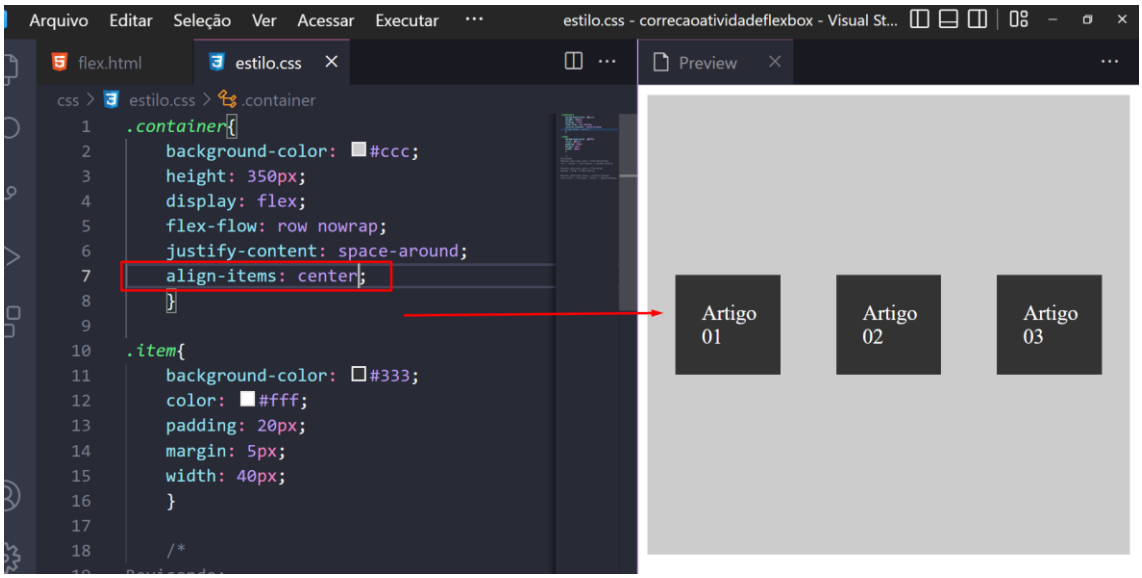
FLEX-END:

Alinha os itens na base do container.



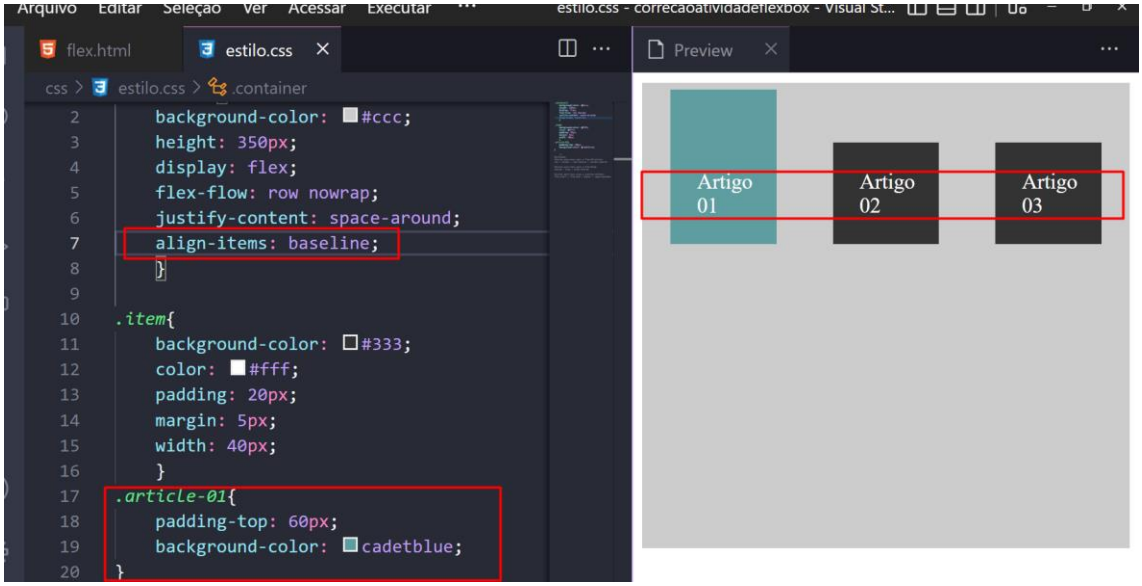
CENTER:

Alinha os itens no centro do container na vertical.



BASELINE:

Esta propriedade faz o alinhamento vertical baseado no conteúdo de cada item. Observe que mesmo o primeiro elemento sendo maior, por conta do padding que foi adicionado, o alinhamento do texto, permaneceu na mesma base, na mesma direção da linha da base.



ALIGN-CONTENT:
O align-content faz um alinhamento vertical quando os itens estiverem em multiline, ou seja, quando houver uma quebra de linha.

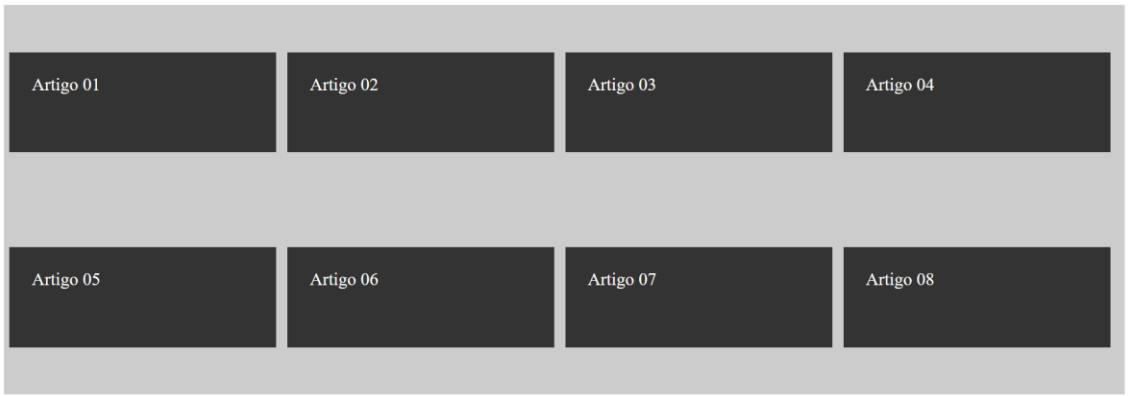
FLEX-START: Alinhamento vertical mesmo com a quebra de linha;

FLEX-END: Alinhamento na base:

CENTER: Alinhamento centralizado

SPACE-BETWEEN: Realiza o espaçamento entre as linhas. Adicionei mais elementos para melhorar a visualização desta formatação.

SPACE-AROUND: Realiza o espaçamento ao redor dos itens.



STRETCH: Estica os elementos. Observação: Para realizar essa ação, é preciso tirar a altura (height) do elemento. Ele é o valor padrão fazendo com que os itens cresçam igualmente na vertical.

Revisando:

Valores possíveis para o Flex-Direction: row | column | row-reverse | column-reverve
Valores possíveis para o Flex-Wrap: nowrap | wrap | wrap-reverse
Valores possíveis para o justify-content: flex-start | flex-end | center | space-between | space-around
Valores possíveis do ALIGN-ITENS: stretch | flex-start | flex-end | center | baseline
Valores possíveis do ALIGN-CONTENT: stretch | flex-start | flex-end | center | space-between | space-around